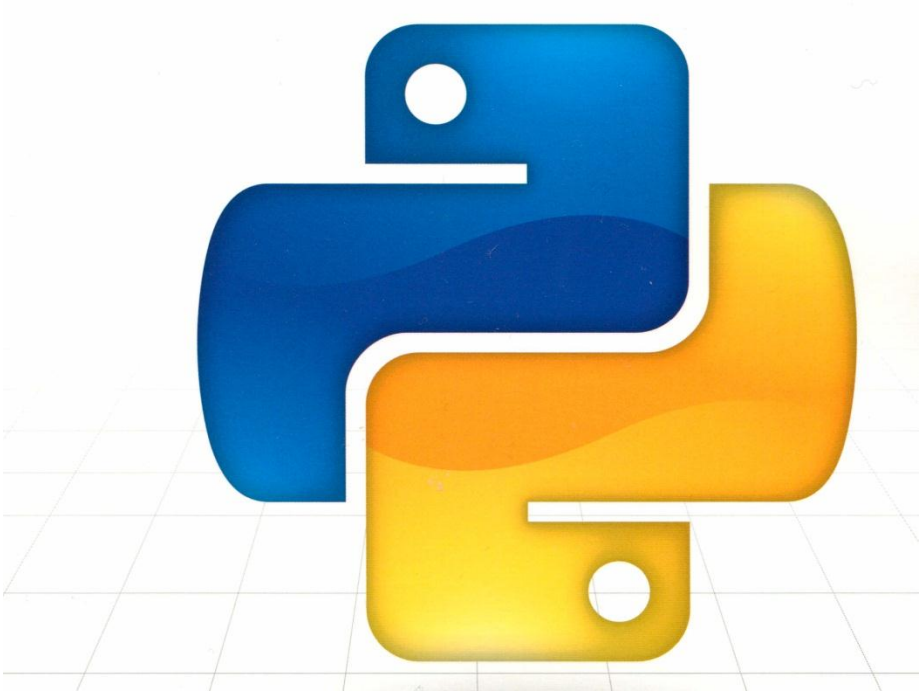
Sh.A. MENGLIYEV, O.A. ABDUG‘ANIEV, S.Q. SHONAZAROV, D. Sh. TO‘RAYEV



PYTHON

Dasturlash tili

Termiz-2021

UDK 681.3.07
BBK 32.973.26

Taqrizchilar: **K.F. Kerimov** – t.f.d.(DSc), dotsent, Toshkent axborot texnologiyalari universiteti “Tizimli va amaliy dasturlashtirish” kafedrasi mudiri;

Ch.B. Normurodov – f.-m.f.d., professor, Termiz davlat universiteti “Amaliy matematika va informatika” kafedrasi mudiri.

Respublikamizda axborot – kommunikatsiya texnologiyalarini keng miqyosda qo‘llashni amalga oshirish yuqori malakali dasturchilarni tayyorlash masalasini ko‘ndalang qo‘ymoqda. Shu maqsadda tayyorlangan ushbu o‘quv qo‘llanma 5330200 – “Informatika va axborot texnologiyalari”(dasturiy ta’minot) ta’lim yo‘nalishi talabalariga “PYTHON dasturlash tili” fanini o‘qitishga mo‘ljallangan, xuddi shuningdek, ushbu o‘quv qo‘llanmadan 5130200 - “Amaliy matematika va informatika”, 5110700 - “Informatika o‘qitish metodikasi” ta’lim yo‘nalishlari va 5A130202 - “Amaliy matematika va axborot texnologiyalari” mutaxasissisligi talabalariga “PYTHON dasturlash tili”, “PHP dasturlash tili”, “Dasturlash tillari”, “Dasturlash asoslari” va “Yuqori bosqichli algoritmik tillar” fanlarini o‘tishga, hamda qo‘llanmadan PYTHON tilida dasturlash ko‘nikmalariga ilgaridan ega bo‘lmagan turli sohada faoliyat yuritayotgan tadqiqotchilar ham foydalanishlari mumkin. Qo‘llanmada hozirgi kunda dunyoda keng miqyosida qo‘llanilayotgan PYTHON dasturlash tili ommabop tarzda bayon qilingan, unda foydali maslahatlar, ko‘plab misol va masalalar, hamda ularning PYTHON tilidagi dasturlari keltirilgan. O‘quv qo‘llanma nafaqat yuqorida ta’kidlangan ta’lim yo‘nalishlari va mutaxassislik talabalari, balki o‘z faoliyatlari doirasida PYTHON tilidan foydalanuvchi tadqiqotchilar hamda tilni mustaqil o‘rganuvchilar uchun foydali manba vazifasini o‘taydi.

Широкое применение информационно – коммуникационных технологий в Республике Узбекистан требует подготовки высококвалифицированных программистов. Подготовленное с этой целью данное учебное пособие предназначено для студентов образовательных направлений образования 5330220 – «Информатика и информационные технологии»(программное обеспечение) по проведенаю предмета «Язык программирования PYTHON», а также данное учебное пособие является полезным источником для студентов по направлениям образование 5130200 – «Прикладная математика и информатика», 5110700 – «Методика обучения информатике» и специальности 5A130202 – «Прикладная математика и информационные технологии» по предметам «Язык программирования PYTHON», «Язык программирования PHP», «Языки программирования», «Основы программирования» и «Алгоритмические языка высокого уровня». Также данное учебное пособие будет полезным для исследователей разных отраслей не имеющих навыков программирования. В учебном пособие популярно изложен язык программирования PYTHON, который широко применяется во всём мире. В пособии даны полезные советы, многочисленные примеры и задачи с кодами для них на языке PYTHON. Также пособие станет полезным источником для исследователей использующих в сфере своей деятельности язык PYTHON и для самостоятельно изучающих данный язык программирования.

The widespread use of information and communication technologies in the Republic of Uzbekistan requires the training of highly qualified programmers. Prepared for this purpose, this tutorial is intended for students in educational areas of education 5330220 - "Informatics and Information Technologies" (software) on the subject "PYTHON programming language ", and this tutorial is a useful source for students in the areas of education 5130200 - "Applied mathematics and computer science ", 5110700 -" Methods of teaching computer science "and specialty 5A130202 -" Applied mathematics and information technology "in the subjects "PYTHON programming language"," PHP programming language "," Programming languages "," Programming basics "and" Algorithmic languages high level". Also, this tutorial will be useful for researchers from different industries who do not have programming skills. The tutorial popularly describes the PYTHON programming language, which is widely used throughout the world. The manual contains useful tips, numerous examples and tasks with codes for them in PYTHON. Also, the manual will become a useful source for researchers using PYTHON in their field and for those who independently study this programming language.

KIRISH

O‘zbekiston Respublikasi Prezidenti Shavkat Mirziyoyev tomonidan 2019 yil yanvar oyida ilgari surilgan beshta muhim tashabbusidan uchinchi tashabbusida aholi va yoshlar o‘rtasida kompyuter texnologiyalari va internetdan samarali foydalanish chora – tadbirlariga oid dasturi doirasida 2019 – 2020 yillarda tuman va shaharlarda raqamli texnologiyalar o‘quv markazi tashkil etish va ularda bepul ta’lim berish, 19 mingga yaqin ijtimoiy soha ob’ektini yuqori tezlikdagi internet tarmog‘iga ulash ko‘zda tutilmoqda.

Hozirgi jadal rivojlanish va turli jarayonlarni avtomatlashtirish hamda robotlashtirish davrida dasturlashni bilish va uni o‘z ish jarayonida ishlata olish texnik va pedagogik yo‘nalishda ta’lim olayotgan o‘quvchi-talabalar uchun juda muhim deb hisoblanadi. Bu zamonaviy mutahassislar uchun eng zaruriy talablardan biridir. Sababi hozirgi kunda informatika turli-tuman sohalarda muvaffaqiyatli ravishda qo‘llanilishi mumkinligini hech kim ham rad eta olmaydi. Huddi shuning uchun ham, o‘quv - qo‘llanmaning asosiy maqsadi – o’quvchi talabalarga Python dasturlash tili misolida hisoblash texnikasi vositalarini ishlatish bo‘yicha bilimlarni va amaliy ko‘nikmalarni imkon darajasida singdirishdir. Amaliy maqsadlarda dasturlash tilining imkoniyatlarini ko‘rsatish matematika va ilmiy sohalarga oid bir qancha misol va masalalarni Python dasturiy tilidan foydalanib yechish misolida aniq va ravshan qilib ko‘rsatib o‘tiladi. Shunday qilib, o‘quv - qo‘llanma o‘quvchi talabalar uchun dasturlash tili vositasida turli xildagi amaliy masalalarni hal qilish ko‘nikmalarini rivojlantirishga imkon yaratadi. O‘quv - qo‘llanmani o‘qish va undagi materiallarni o‘rganish uchun dasturlash tajribasi bo‘lishi talab etilmaydi va undan endigina dasturchi bo‘lishni orzu qilganlar ham bemalol foydalanishlari mumkin. Shuni ham ta’kidlash kerakki, ushbu o‘quv - qo‘llanmada zamonaviy Python dasturlash tili imkoniyatlari boshlang‘ich o‘rganuvchilar, ya’ni maktab o‘quvchilari, talabalar va mustaqil o‘rganuvchilar tushunishi uchun nihoyatda yengil, tushunarli, kerakli izohlar bilan va sodda xalq tilda izhor qilingan. Python dasturlash tili samarador yuqori darajadagi ma’lumotlar tuzilmasini hamda oddiy, ammo samarador bo‘lgan ob’yektga yo‘naltirilgan dasturlash uslublarini taqdim etadi. Undan tashqari, bu til o‘rganish uchun oson va shu bilan birga imkoniyatlari yuqori bo‘lgan oz sonli dasturlash tillari jumlasiga kiradi va shu bilan birgalikda unda dasturlash jarayoni juda ham oddiy amalga oshiriladi. Python dasturlash tilining rasmiy sayti – www.python.org bo‘lib, uning muallifi Niderlandiyadagi Matematika va informatika ilmiy adqiqot institutida ishlagan *Gvido van Rossum* deb hisoblanadi. Pythonning o‘ziga xosligi esa uning oddiyligi, o‘rganishga osonligi, sodda sintaksisga egaligi va dasturlash jarayonini boshlash uchun qulay, erkin va ochiq kodlik dasturiy ta’minotga egaligidir. Undan tashqari, o‘z dasturingizni yozish davomida quyi darajadagi detallarni, misol uchun xotirani boshqarishni hisobga olishingizga hech qanday hojat qolmaydi. Bu dasturlash tili ko‘plab platformalarda hech qanday o‘zgartirishlarsiz ishlay oladi va u interpretatsiya qilinadigan tillar jumlasiga mansub.

Bulardan tashqari, Python dasturlash tili imkoniyatlari kengayishga moyil bo‘lgan dasturiy til hisoblanadi. Agar siz dasturingizning biror-bir joyini tezroq ishlashini xoxlasangiz, o‘sha qismni C yoki C++ dasturlash tillarida yozib, keyin shu qismni Python kodingiz orqali ishga tushirsangiz (chaqirsangiz) bo‘ladi. Bundan tashqari, Python juda ham ko‘p, foydali hamda xilma-xil dasturlar kutubxonalarga egaligi ham juda muhimdir. Python dasturlash tili sodda va o‘qilishi oddiy bo‘lgan dasturlash tili bo‘lib u inglizcha so‘zlarni qo‘llaydi va u PERL va PHP ga tillariga o‘xshab ketadi. Python interaktiv dasturlash tili bo‘lib, ob’ektga yo‘naltirilgan tillar jumlasiga kiradi, ya’ni, Python ob’ektga yo’naltirish uslubini yoki dasturiy texnikasini qo‘llab-quvvatlaydi. Python boshlovchi dasturchilar tilidir, ya’ni u boshlang‘ich dasturchilar uchun ajoyib til bo‘lib, oddiy matnni ishlashdan tortib, veb-brauzerlaridagi o‘yinlarga qadar keng ko‘lamdagi ilovalarni ishlab chiqishni qo‘llab quvvatlaydi. Python ning buyruqlari va sintaksisi ABC, Modula-3, C, C++, Algol-68, SmallTalk va Unix shell kabi boshqa ko‘plab tillardan va skript tillaridan olingan. Python mualliflik huquqi bilan himoyalangan. Xuddi Perl kabi, Python dagi manbaa kodi GNU General Public License (GPL) ostida mavjud. Pythonning o‘ziga xos xususiyatlari quyidagilarni o‘z ichiga qamrab oladi:

- *O‘rganish oson*: Python nisbatan kam sonli kalit so‘zlar, oddiy tuzilish va aniq belgilangan sintaksisga ega;
- *Tushunish va o‘qish oson*: Python kodi juda aniq va yodda qoladigan tarzda yoziladi;
- *Unda ishlash juda ham qulay*: Python ning muvaffaqiyati – manba kodining tuzilishi juda sodda va tushunarli;
- *Python kattagina standart kutubxonaga ega*: Python ning eng qudratli jihatlaridan biri kutubxonaning asosiy qismi juda portative va UNIX, Windows va Macintosh-da o‘zaro faoliyat platformalar bilan mos keladi;
- *Interaktiv usulda ishlash imkoniayti mavjud*: Python da terminalda ishlash uchun juda qulay, natijalarni terminalda test qilib ko‘rsa ham bo‘ladi;
- *Bu til moslashuvchan hisoblanadi*: Python keng apparat platformalarida ishlaydi va barcha platformalarda bir xil interfeysga ega;
- *Kengaytirilish imkoniyatalariga ega*: Python tarjimoniga past darajadagi modullarni qo‘shishingiz mumkin;
- *Ma’lumotlar bazalari bilan ishlash qulayligi*: Python barcha a’lumotlar bazasini qo‘llab quvvatlaydi;
- *GUI dasturlashni amalga oshirish imkoniyati*: Python Windows MFC, Unix, X Window kabi platformalarga GUI dasturlar tuzishni qo’llab quvvatlaydi;
- *Moslashuvchanligi*: Python qobiq buyruq fayliga qaraganda, katta dasturlarga yanada yaxshi moslashish va ularni qo‘llab-quvvatlash imkonini beradi;
- *Funktsional va tuzilgan dasturiy usullarni* va Ob’ektga yo‘naltirilgan dasturlashni qo‘llab-quvvatlaydi;
- *Buyruq fayli sifatida ishlatilishi mumkin* yoki katta ilovalar yaratish uchun byte-kodga to‘planishi mumkin;

- *Juda yuqori darajadagi dinamik ma'lumotlar turlari* va dinamik turdagi tekshiruvlarni qo‘llab-quvvatlaydi;
- Chiqindilarni avtomatik ravishda to‘plashni va ularni tozalashni qo‘llab-quvvatlaydi (*musorosborshik funktsiyasi*);
- C, C++, Java va PHP kabi dasturlash tillari bilan osonlik bilan bog‘lanishi mumkin.

Python dasturlash tili boshqa tillarga nisbatan o‘rganish ancha oson va shu bilan birga imkoniyatlari boy bo‘lgan til hisoblanadi. Ya'ni, til o‘rganishni boshlovchilar uni osonlik bilan o‘rganishlari mumkin, shu bilan bu til yordamida ancha-muncha jiddiy amaliy loyihalarni ham amalga oshirish mumkin.

Python haqida quyidagi uchta xulosaga kelish mumkin:

1. Python dasturlash tilining keng miqyosda qo‘llanilishi mumkin bo‘lgan uch asosiy soha bor: veb-dasturlash (*backend – vebserver uchun ilovalar yozish*), sun'iy intellekt masalalari, kompyuterda foydalanuvchi juda ko‘p marta bajaradigan mayda ishlar (*elektron xatlarni jo'natish, fayllarni izlash va bosmalash, elektron jadvaldan biror-bir ma'lumotlarni ajratib olish va xakozolar*).

2. Python o‘rganish ancha oson bo‘lgan dasturiy tildir. Agar tabiiy tillar bilan o‘xshatish qiladigan bo‘lsak, biror-bir tilda fikrni yetkazish uchun ma'lum vaqt so‘zlarni, tilning grammatikasi o‘rganish kerak bo‘ladi. Qandaydir minimal bilim shakllangandan so‘ng, asta-sekin inson o‘z fikrini ifoda eta boshlaydi. Dasturlash tillari bilan ham holat xuddi shunday. Biror dasturlash tilida amaliy foyda keltiradigan dastur yozishni boshlash uchun ma'lum bilimlar majmuini egallash kerak, shundan so‘nggina dasturlashni boshlash mumkin. Boshqa dasturlash tillaridan farqli ravishda, Python da amaliy ahamiyatga ega dasturlarni ishlab chiqishga ancha ertaroq, hali tilning katta qismini o‘rganmasdan turib ham kirishish mumkin.

3. Python interpretatsiya qilinadigan dasturiy til. Dasturlash tillarini interpretatsiya qilinadigan va kompilyatsiya qilinadigan dasturlash tillariga bo‘lishadi. Aniqroq aytganda, agar dasturlash tilidagi dasturni bajarish interpretatsiya orqali amalga oshirilsa, bunday tillar interpretatsiya qilanadigan til deyiladi. Agar dasturlash tilidagi dasturni bajarish uchun uni avval mashina tiliga o‘tkazish talab qilinsa, bunday tillar kompilyatsiya qilinadigan tillar deyiladi. Aslini olganda, kompyuter uchun yozilgan har qanday dastur interpretatsiya qilinadi. Chunki mashina kodlaridagi dastur kompyuterning miyasi bo‘lgan protsessor tomonidan interpretatsiya qilinadi. Interpretatsiya qilinadigan tillarda yozilgan dasturlar uchun maxsus – interpretator dastur mavjud. Bu interpretator dastur kodlarini bajarilishini ta'minlab beradi. Bu o‘quv - qo‘llanma dasturlashni o‘rganuvchilar hamda ilmiy yoki amaliy maqsadlarni amalga oshirish uchun bu dasturlash tilini o‘rganishi kerak bo‘lgan insonlar uchun mo‘ljallangan. Ushbu qo‘llanmaning asosiy maqsadi - Sizga Python tilida dasturlashning nazariy va amaliy asoslarini o‘rgatishdan iboratdir. Dasturlash tilini o‘rganish uchun eng asosiy amal – kitobda berilgan barcha topshiriqlarni o‘z vaqtida, tushungan holda va

aniq bajarishdir. Chunki, har qanday soha bo‘yicha chuqur bilim faqatgina amaliyot orqali puxta egallanadi.

PYTHON – dasturlash tili bo‘yicha o‘zbek tilidagi adabiyotlar yetarli darajada emasligi, ko‘pgina foydalanuvchilarning ushbu tilda dastur tuzishlariga to‘sqinlik qilmoqda. Shu sababli, keng doiradagi foydalanuvchilarga mo‘ljallangan, tushunarli tilda yozilgan o‘quv qo‘llanmalarga bo‘lgan ehtiyoj kundan-kunga ortib bormoqda.

Ushbu o‘quv qo‘llanma Termiz davlat universiteti “Amaliy matematika va informatika”, “Informatika o‘qitish metodikasi” ta’lim yo‘nalishlari, hamda “Amaliy matematika va axborot texnologiyalari” mutaxasissisligi talabalariga “Dasturlash tillari”, “Dasturlash asoslari” va “Yuqori bosqichli algoritmik tillar” fanilarida “PYTHON dasturlash tili” ni o‘tishga mo‘ljallab yozilgan bo‘lib, unda ko‘pgina amaliy xarakterga ega bo‘lgan – dasturiy kodlar va misollarning PYTHON tilida tugallangan dasturlari keltirilgan. O‘quv qo‘llanma uni o‘zlashtirish uchun maxsus bilimlarni talab qilmaydigan ketma – ketlikda bayon qilingan.

Mazkur o‘quv qo‘llanma oltita bobdan iborat bo‘lib:

- I BOB. PYTHON TILI VA UNING DASTURLASH MUHITI
- II BOB. PYTHON DA OPERATORLAR VA ULAR BILAN ISHLASH
- III BOB. PYTHON DA TARMOQLANUVCHI OPERATORLAR
- IV BOB. PYTHON DA TAKRORLASH OPERATORLARI
- V BOB. PYTHON DA MASSIVLAR
- VI BOB. PYTHON DA FUNKSIYALAR
- VII. BOB. PYTHON DA MATNLAR BILAN ISHLASH

Har bir bob yakunida 20 tadan misol va masalalarning PYTHON tilidagi dasturlari keltirilgan hamda talabalar mustaqil ishlashlari uchun 20 tadan topshiriqlar berilgan.

I. BOB. PYTHON TILI VA UNING DASTURLASH MUHITI

1.1 PYTHON TILI TARIXI

Python dasturlash tilini yaratilishi 1990-yil boshlaridan boshlangan. O‘sha paytlarda uncha taniqli bo`lmagan Gollandiyaning CWI institute xodimi Gvido van Rossum ABC tilini yaratilish proektida ishtirok etgan edi. ABCtili Basic tili o‘rniga talabalarga asosiy dasturlash konsepsiyalarini o‘rgatish uchun mo`ljallangan til edi. Bir kun Gvido bu ishlardan charchadi va 2 hafta davomida o‘zining Macintoshida boshqa oddiy tilning interpretatorini yozdi, bunda u albatta ABC tilining ba’zi bir g‘oyalarini o‘zlashtirdi. Shuningdek, Python 1980-1990-yillarda keng foydalanilgan Algol-68, C, C++, Modul3 ABC, SmallTalk tillarining ko‘plab xususiyatlarini o‘ziga olgandi. Gvido van Rossum bu tilni internet orqali tarqata boshladi. Bu paytda o‘zining “Dasturlash tillarining qiyosiy taqrizi” veb sahifasi bilan internetda to 1996-yilgacha Stiv Mayevskiy ismli kishi taniqli edi. U ham Macintoshni yoqtirardi va bu narsa uni Gvido bilan yaqinlashtirdi. O‘sha paytlarda Gvido BBC ning “Monti Paytonning havo sirki” komediyasining muxlisi edi va o‘zi yaratgan tilni Monti Payton nomiga Python deb atadi (ilon nomiga emas). Til tezda ommalashdi. Bu dasturlash tiliga qiziqqan va tushunadigan foydalanuvchilar soni ko‘paydi. Boshida bu juda oddiy til edi. Shunchaki kichik interpretator bir nechta funksiyalarga ega edi. 1991-yil birinchi OYD(Obyektga Yo‘naltirilgan Dasturlash) vositalari paydo bo‘ldi. Bir qancha vaqt o‘tib Gvido Gollandiyadan Amerikaga ko‘chib o‘tdi. Uni NRI korparatsiyasiga ishlashga taklif etishdi. U o‘sha yerda ishladi va korparatsiya shug‘ullanayotgan proektlarni Python tilida yozdi va bo‘sh ish vaqtlarida tilni interpretatorini rivojlantirib bordi. Bu 1990-yil Python 1.5.2 versiyasi paydo bo‘lguncha davom etdi. Gvidoning asosiy vaqti korparatsiyani proektlarini yaratishga ketardi bu esa unga yoqmasdi. Chunki uning Python dasturlash tilini rivojlantirishga vaqti qolmayotgandi. Shunda u o‘ziga tilni rivojlantirishga imkoniyat yaratib bera oladigan homiy izladi va uni o‘sha paytlarda endi tashkil etilgan BeOpen firmasi qo‘llab quvvatladi. U CNRI dan ketdi, lekin shartnomaga 8 binoan u Python 1.6 versiyasini chiqarib berishga majbur edi. BeOpen da esa u Python 2.0 versiyani chiqardi. 2.0 versiyasi bu oldinga qo‘yilgan katta qadamlardan edi. Bu versiyada eng asosiysi til va interpretatorni rivojlanish jarayoni ochiq ravishda bo‘ldi. Shunday qilib 1.0 versiyasi 1994-yil chiqarilgan bo‘lsa, 2.0 versiyasi 2000- yil, 3.0 versiyasi esa 2008-yil ishlab chiqarildi. Hozirgi vaqtda uchinchi versiyasi keng qo‘llaniladi.

Python dasturlash tili imkoniyatlari Python – bu o‘rganishga oson va shu bilan birga imkoniyatlari yuqori bo‘lgan oz sonlik zamonaviy dasturlash tillari qatoriga kiradi. Python yuqori darajadagi ma‘lumotlar strukturasi va oddiy lekin samarador obyektga yo'naltirilgan dasturlash uslublarini taqdim etadi.

Pythonning oʻziga xosligi

1. Oddiy, oʻrganishga oson, sodda sintaksisga ega, dasturlashni boshlash uchun qulay, erkin va ochiq kodlik dasturiy taʼminot.

2. Dasturni yozish davomida quyi darajadagi detallarni, misol uchun xotiraniboshqarishni hisobga olish shart emas.

3. Koʻplab platformalarda hech qanday oʻzgartirishlarsiz ishlay oladi.

4.Interpretatsiya qilinadigan til.

5. Kengayishga moyil til. Agar dasturni biror joyini tezroq ishlashini xoxlasak shu qismni C yoki C++ dasturlash tillarida yozib keyin shu qismni python kodi orqali ishga tushirsa(chaqirsa) bo'ladi.

6.Juda ham ko'p xilma-xil kutubxonalarga ega.

7.xml/html fayllar bilan ishlash

8.http so`rovlari bilan ishlash

9. GUI(grafik interfeys)

10.Veb saytlarni yaratish

11.FTP bilan ishlash

12. Rasmli audio video fayllar bilan ishlash

13.Robot texnikada

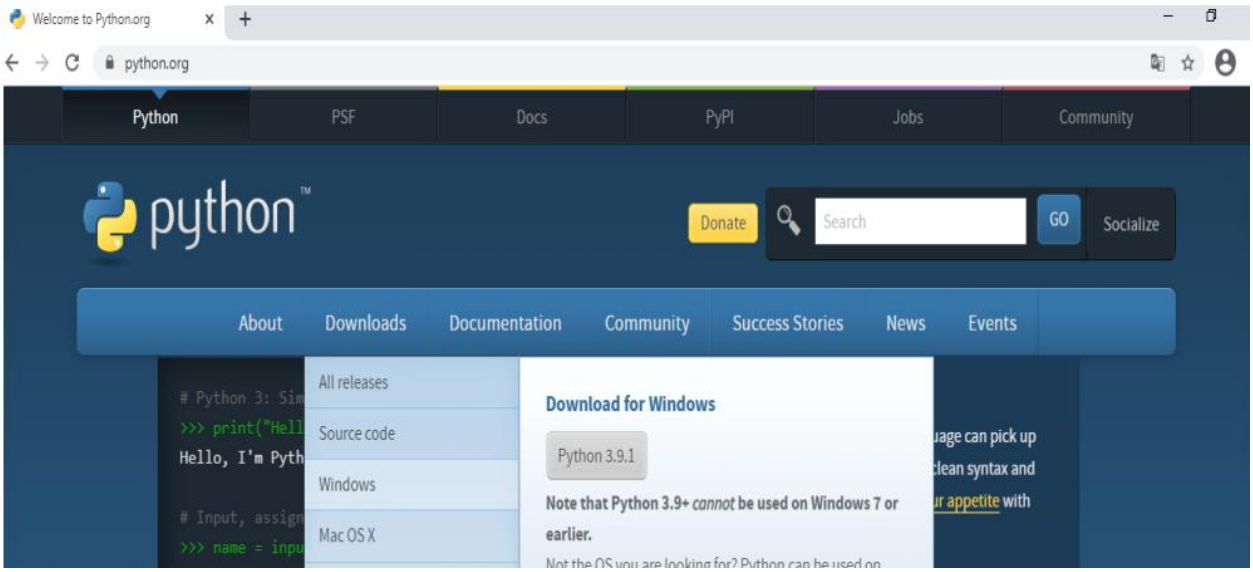
14.Matematik va ilmiy hisoblashlarni dasturlash

Pythonni katta proyektlarda ishlatish mumkin. Chunki, uni chegarasi yoʻq, imkoniyati yuqori. Shuningdek, u sodda va universalligi bilan dasturlash tillari orasida eng yaxshisidir.

1.2. PYTHON DASTURINI OʻRNATISH QOIDALARI

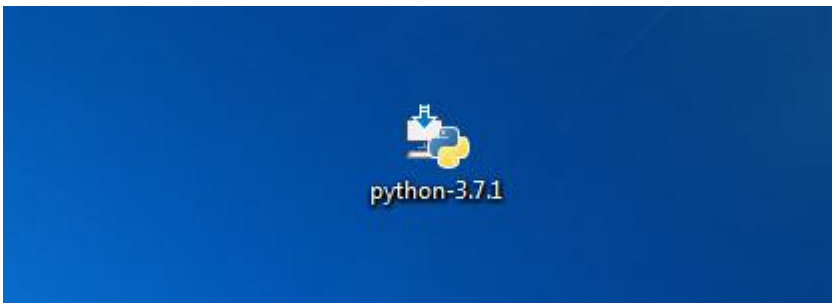
Python dasturini kompyuterga oʻrnatish bir necha bosqichlardan iborat.

1. https://www.python.org/downloads/windows/ orqali rasmiy veb saytiga kirib, kompyuteringizning texnik parametrlarini hisobga olgan holda eng soʻnggi versiyasini yuklab olamiz. Kompyuterning texnik parametrlariga, razryadi, (64 bit yoki 32 bit) qaysi operatsion sistema oʻrnatilganligi va boshqa parametrlar kiradi. Biz hozir sizga 64 bitli kompyuter uchun oʻrnatish jarayonini tushuntirib oʻtamiz.



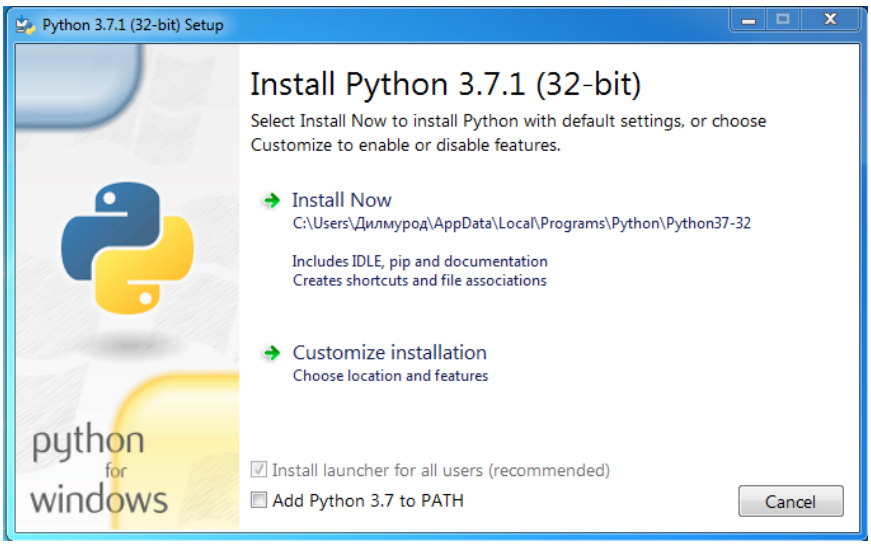
1-Rasm

2. Yuklab olib o‘rnatishni boshlaymiz. Pastdagi rasmlarda o‘rnatish jarayoni rasmlar ketma – ketligida keltirilgan.



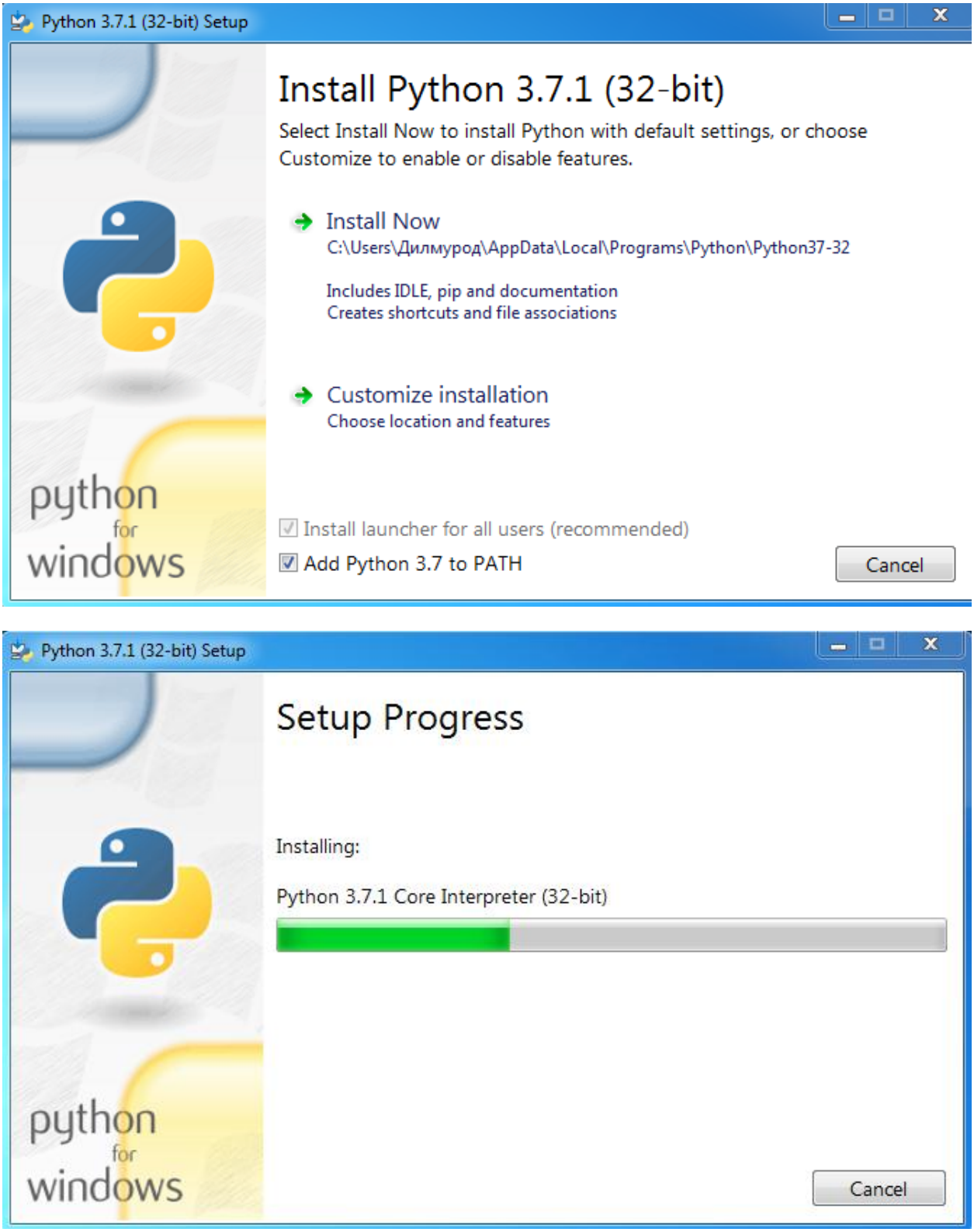
2-Rasm

Dastur ustiga sichqonchani ikki marta bosib, o‘rnatishni boshlaymiz. Quyidagi oyna hosil bo‘ladi.



3-Rasm

Bu oynadan Add Python 3.7 to Path ga belgi qo‘yib, Install Now ni tanlaymiz.



4-Rasm

Dastur o‘rnatib bo‘lingach Closeni bosib, ishni yakunlaymiz.

5-Rasm

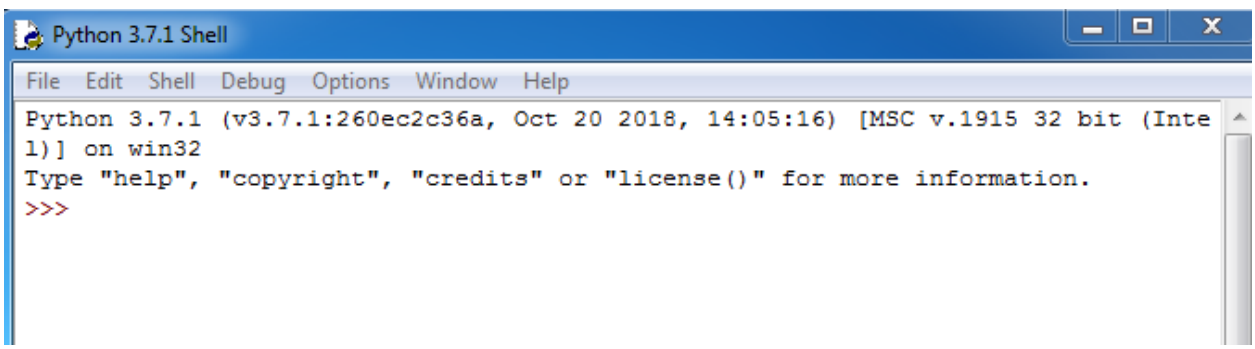
Shuni alohida ta'kidlash kerakki, dastur o'rnatib bo'lingach, Python dasturlash tilining ishchi stoliga alohida belgisi hosil bo'lmaydi. Shuning uchun dasturni Pusk orqali ishga tushiramiz. Har doim yangi dastur bilan ishlashda IDLE ni ishga tushiramiz.

1.3. IDLE ni ishga tushirish tartibi

Har doim yangi dastur tuzishda IDLE alohida ishga tushiriladi, ishga tushirish tartibi esa doim bir xil ko'rinishda bo'ladi.

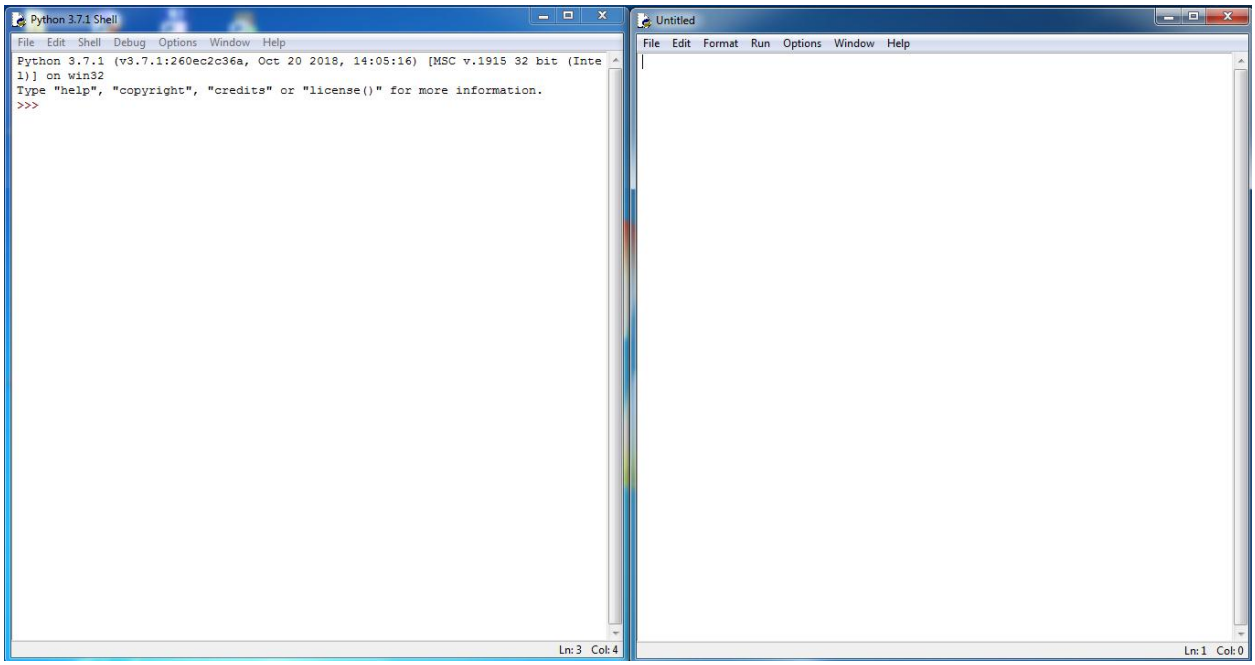
1.Puskdan Python 3.7 ni topamiz va sichqonchani Python 3.7 ustiga bosib IDLE ni tanlaymiz.

2.Yangi oq oyna hosil bo'ladi.



6-Rasm

3.File bo’limidan New File ni tanlab (klaviaturadan Ctrl+N), yangi ikkinchi oq oynani hosil qilamiz.



7-Rasm

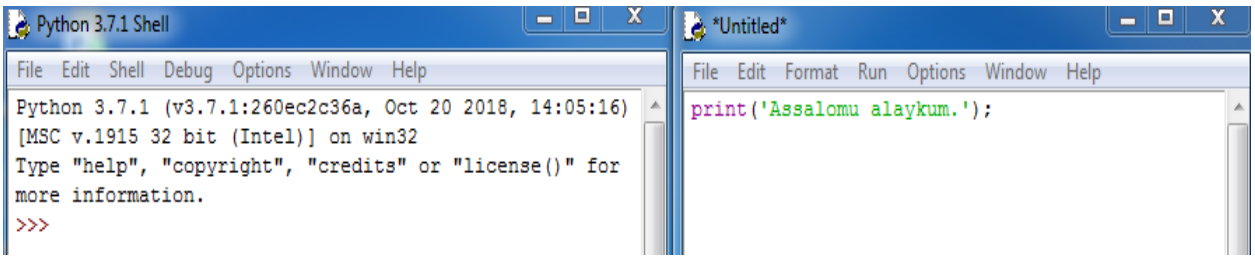
Ikkinchi oyna kod yozish uchun, birinchi oyna esa dastur natijasini ko‘rish uchun ishlatiladi. Unutmang, ikkinchi oynada kodlarni yozib bo‘lgach uni saqlab olishimiz kerak, aks holda dastur ishlamaydi. Saqlash uchun ishchi stolidan “Dasturlar” nomli papka hosil qilib, hamma dasturlarni shu papkaga saqlaymiz. Hozir namuna sifatida biror dastur yozib, uni saqlab ishga tushirishni o‘rganamiz.

1.Ishchi stolidan “Dasturlar” nomli papka yaratamiz.

8-Rasm

1.IDLE ni yuqoridagi tartibda ishga tushiramiz. Ikkinchi oynaga (oxirgi ochilgan oynaga) namunaviy kodlarni yozamiz (bundan keyin kodlarni yozamiz deyilganda ikkinchi oyna nazarda tutiladi.).

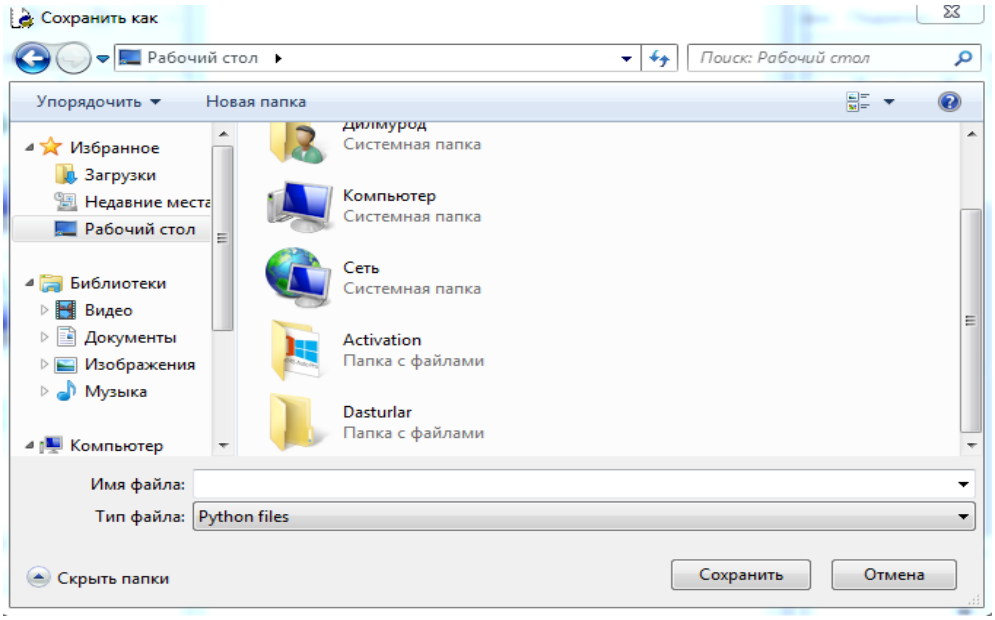
2.Eslatma: dastur kodlarini yozayotganda o‘,g‘ harflaridan foydalanish noqulayliklar tug‘diradi. Shuning uchun bu harflardan foydalanmaymiz.



9-Rasm

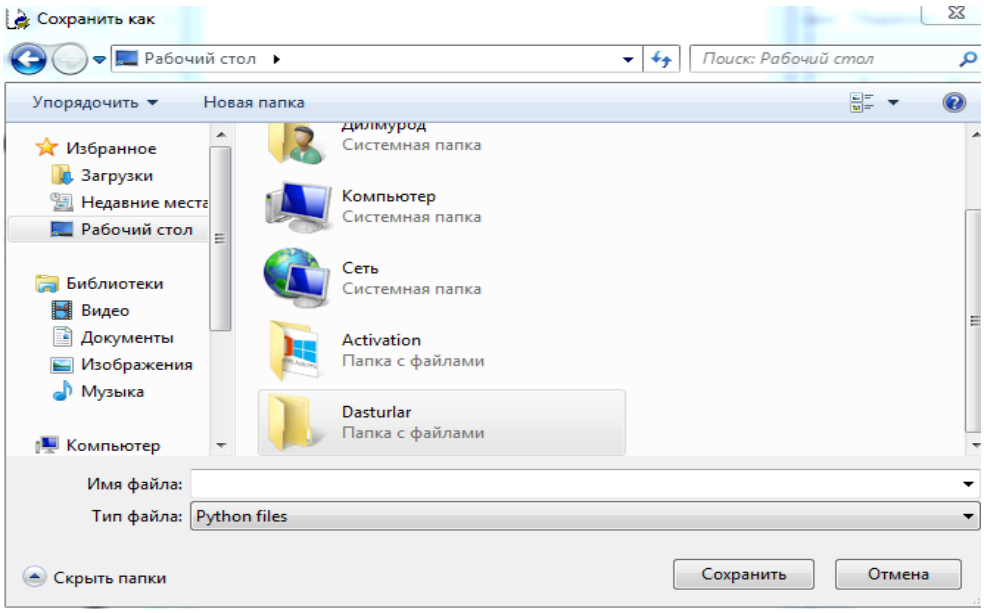
3.File bo'limidan Save as ni tanlaymiz.

4.Ochilgan oynadan ishchi stolini tanlaymiz.



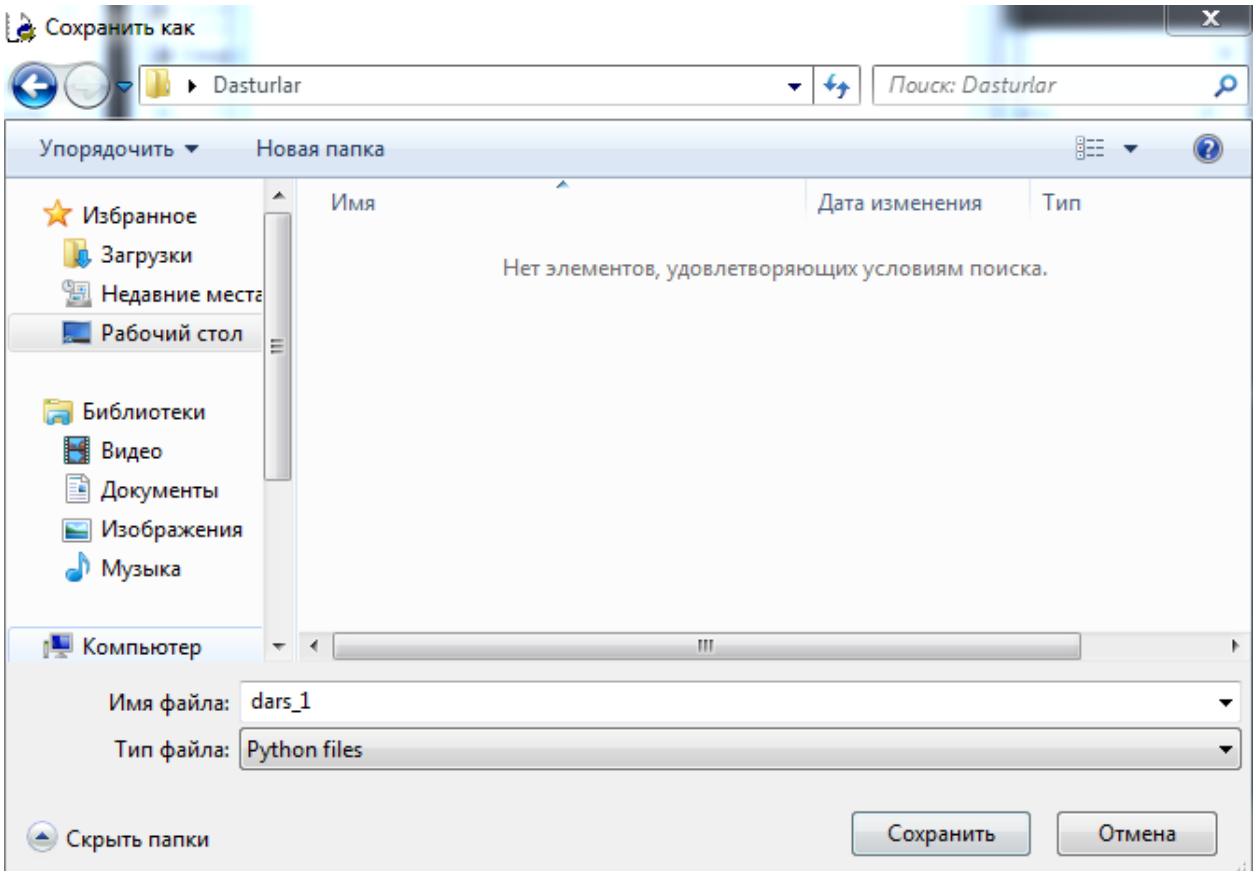
10-Rasm

5.Yangi yaratgan Dasturlar nomli papkani tanlaymiz.



11-Rasm

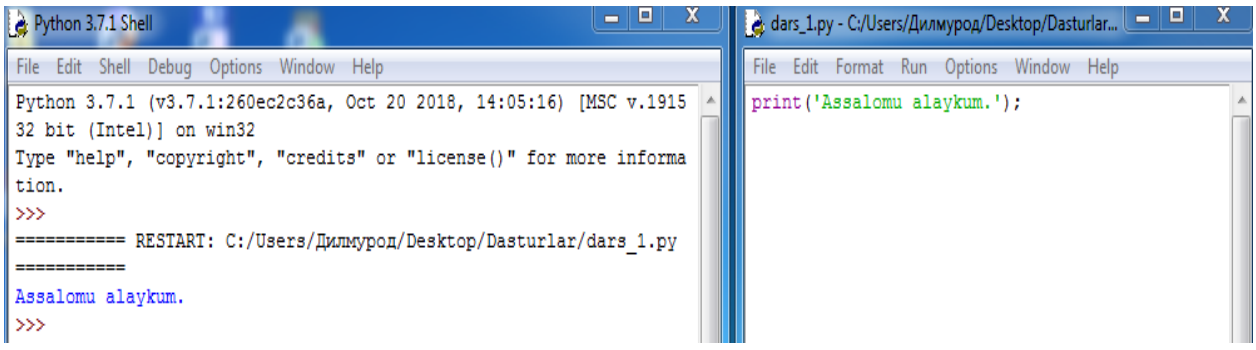
6.Dasturga nom berib saqlaymiz. (masalan dars_1). Nom berishda joy tashlamaslik kerak, uning o‘rniga tag chiziq (_) dan foydalaning.



12-Rasm

7.Dasturni tekshirish uchun, Run bo‘limidan Run Module ni tanlaymiz.

8.Natija birinchi oq oynada hosil bo‘ladi. (Natija har doim birinchi oynada ko‘rsatiladi.)



13-Rasm

Dasturimiz natija berdi, demak kodlarni to‘g‘ri kiritdik. Agar yozilgan dasturga biror o‘zgartirish kiritsak klaviaturadan Ctrl+S klavishlar birikmasini bosgan holda saqlab, yana Run bo’limi orqali qayta ishga tushirish mumkin.

Python dasturida ishlash tartibi quyidagicha ekan:

1.Puskdan Python 3.7 ni topib, undan IDLEni ishga tushirish

2.Ochilgan oynaning File bo‘limidan New File ni tanlash (File →New File)

3.Yangi ochilgan oynaga kodlarni kiritish va saqlash (ishchi stolida yaratilgan yangi papkaga, masalan Dasturlar nomli papkaga) (File → Save as →Ishchi stoli (Рабочи стол) →Yangi yaratilgan papka (Dasturlar) →Dastur nomi (dars_1) →Сохранить (saqlash))

4. Dasturni ishga tushirish (Run →Run Module)

5.Natijani tekshirish 6.Agar dasturga biror o‘zgartirish kiritilsa klaviaturdan CTRL+S klavish birikmalari yordamida qayta saqlash.

7.Har doim yangi dastur yaratilayotganda yuqoridagi tartiblar takroran bajariladi.

8.Keyingi darslarda dasturni ishga tushiramiz deyilganda yuqoridagi tartiblar tushuniladi.

9.Ma’lumotni har safar kiritayotganimizda (dastur natijasini tekshirishda) enter klavishini bosamiz.

1.4 PYTHON da kiritish va chiqarish operatorlari

Ma’lumotni konsol ekraniga chiqarish – *print()* funksiyasi hisoblanadi.
Funksiyaga argument sifatida konsolga chiqariluvchi qiymatlar (satr, son, ifoda va x.k.) berilishi mumkin:

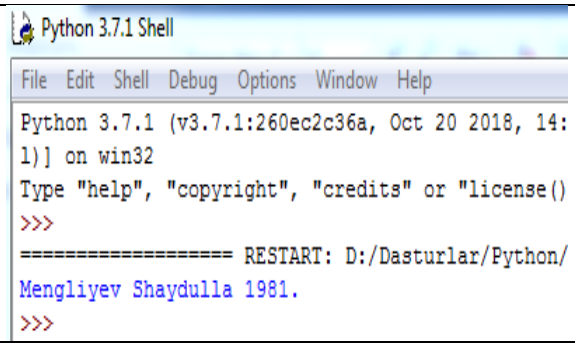
print('Hello world!')	Python 3.7.1 Shell File Edit Shell Debug Options Window Help Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14: 1)] on win32 Type "help", "copyright", "credits" or "license() >>> ========== RESTART: D:/Dasturlar/Python/ Hello world! >>>

Agarda birdaniga bir nechta qiymatlarni chop etish talab qilinsa, u holda ularni *print()* funksiyasiga “,” bilan ajratib kiritiladi:

Familiya='Mengliyev'; Ism='Shaydulla'; Tugilgan_yili=1981; print(Familiya,Ism,Tugilgan_yili);	Python 3.7.1 Shell File Edit Shell Debug Options Window Help Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14: 1)] on win32 Type "help", "copyright", "credits" or "license() >>> ========== RESTART: D:/Dasturlar/Python/ Mengliyev Shaydulla 1981 >>>

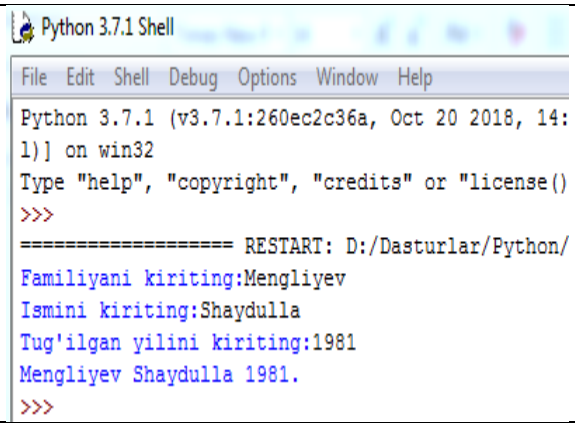
Ma’lumotlar ekranga chiqqanida ma’lumotning oxiriga nuqta, vergul, probel va h.k. belgilarni chiqarish uchun *print()* ning *end* xususiyatidan foydalanamiz:

```
Familiya='Mengliyev';
Ism='Shaydulla';
Tugilgan_yili=1981;
print(Familiya,Ism,Tugilgan_yili,end='.');
```



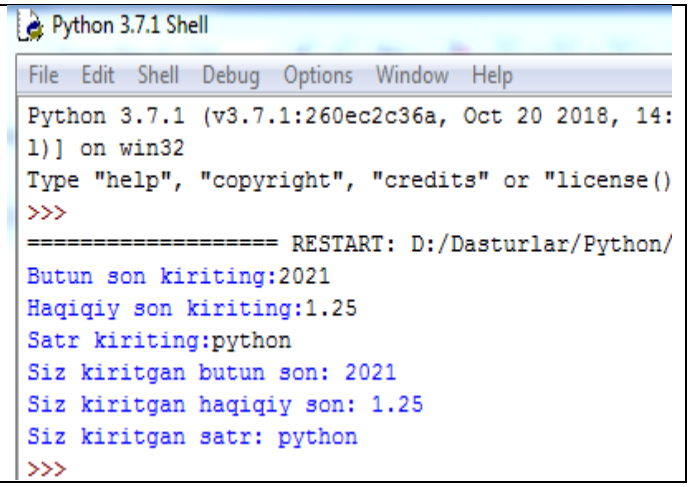
input() ekrandan berilganlarni kiritish uchun qo‘llaniladi. input() funksiyasiga argument sifatida biror bir satr berilishi mumkin. Ushbu satr konsol ekranida aks ettirilib, kiritilishi kerak bo‘lgan berilganlar uchun yordamchi taklif vazifasini bajaradi. Masalan:

```
Familiya=input('Familiyani kiriting:');
Ism=input('Ismini kiriting:');
Tugilgan_yili=input("Tug'ilgan yilini
kiriting:");
print(Familiya,Ism,Tugilgan_yili,end='.');
```



Kiritilayotgan ma’lumotlarni aniq biror bir turga tegishli qilish mumkin:

```
butun_son=int(input('Butun son
kiriting:'));
haqiqiy_son=float(input('Haqiqiy son
kiriting:'));
satr=str(input('Satr kiriting:'));
print("Siz kiritgan butun
son:",butun_son);
print("Siz kiritgan haqiqiy
son:",haqiqiy_son);
print("Siz kiritgan satr:",satr);
```



1.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Masala 1. PYTHON da o‘zingizning “**Familiya va Ismingizni**” ekranga chiqaring.
Masala 2. PYTHON da o‘zingizning “**Yo’nalishingizni**” ekranga chiqaring.
Masala 3. PYTHON da o‘zingizning “**Universitetingizni nomini**” ekranga chiqaring.
Masala 4. PYTHON da o‘zingizning “**Fakultetingiz nomini**” ekranga chiqaring.
Masala 5. PYTHON da o‘zingizning “**Guruhingiz nomini**” ekranga chiqaring.
Masala 6. PYTHON da ushbu gapni “**O’zbekiston kelajagi buyuk davlat!**” ekranga chiqaring.
Masala 7. PYTHON da ushbu gapni “**Men PYTHON dasturlash tilini o‘rganmoq-chiman**” ekranga chiqaring.
Masala 8. PYTHON da **ax+b=0** ifodani kiriting va ekranga chiqaring.
Masala 9. PYTHON da **sinx+cosx=1** ifodani kiriting va ekranga chiqaring.
Masala 10. PYTHON da **tgx+ctgx=1** ifodani kiriting va ekranga chiqaring.
Masala 11. PYTHON da **sinx+ctgx=1+cosx** ifodani kiriting va ekranga chiqaring.
Masala 12. PYTHON da **ax-by-hz-n=0** ifodani kiriting va ekranga chiqaring.
Masala 13. PYTHON da O‘zbekinton Respublikasi madhiyasining birinchi to‘rtligini ekranga chiqaring.
Masala 14. PYTHON da O‘zbekinton Respubliksi madhiyasining ikkinchi to‘rtligini ekranga chiqaring.
Masala 15. PYTHON da O‘zbekinton Respubliksi madhiyasining uchinchi to‘rtligini ekranga chiqaring.
Masala 16. PYTHON da **ax+by-1=0** ifodani kiriting va ekranga chiqaring.
Masala 17. PYTHON da **bx=1-k** ifodani kiriting va ekranga chiqaring.
Masala 18. PYTHON da **ax+bx=cz-dk** ifodani kiriting va ekranga chiqaring.
Masala 19. PYTHON da **ax+bx-dy=8** ifodani kiriting va ekranga chiqaring.
Masala 20. PYTHON da **ax+bx+ck=10** ifodani kiriting ekranga chiqaring.

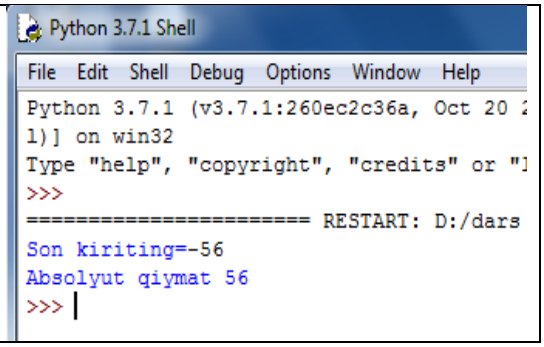
II.BOB. PYTHON DA OPERATORLAR VA ULAR BILAN ISHLASH

2.1. PYTHON DA MATEMATIK FUNKSIYALAR

Pythonning matematik funksiyalar kutubxonasi trigonometrik hisoblashlar, sonli shakl almashtirishlar va sonli almashtirishlarni bajaradi. Trigonometrik funksiyalar argumentlari radianlarda beriladi, hamda graduslarni radianga va aksincha almashtiruvchi funksiyalar ham mavjud. Matematik operatorlar bilan bir qatorda Pythonda ko‘p sonli matematik funksiyalar ham nazarda tutilgan.
Bular quyidagilar:

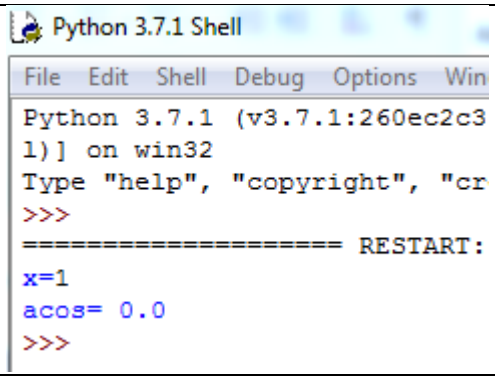
- abs() - sonning absolyut qiymati.

```
a = int(input('Son kiriting='))
Absolyut_qiymat = abs(a)
print('Absolyut qiymat',Absolyut_qiymat)
```



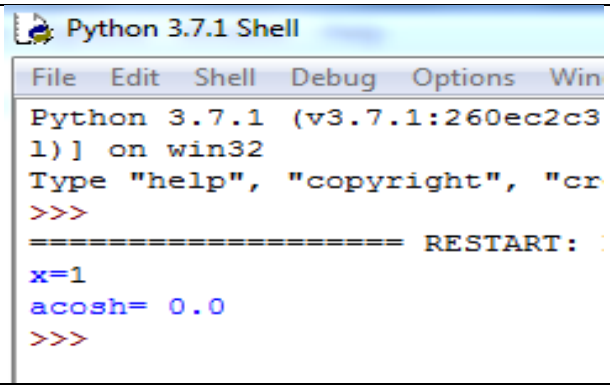
- acos() – radianda ifodalangan arkkosinus.

```
import math
x=float(input('x='))
y=math.acos(x)
print('acos=',y)
```



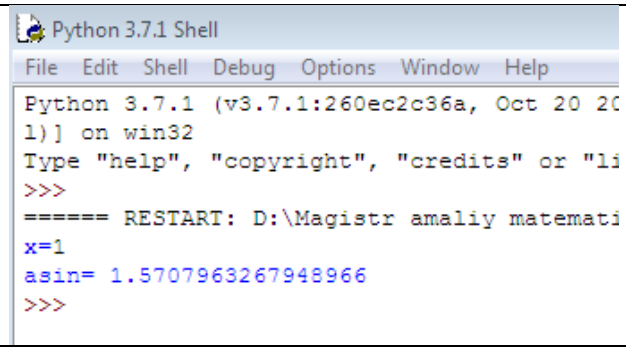
- acosh() - radianda ifodalangan giperbolik arkkosinus.

```
import math
x=int(input('x='))
y=math.acosh(x)
print('acosh=',y)
```



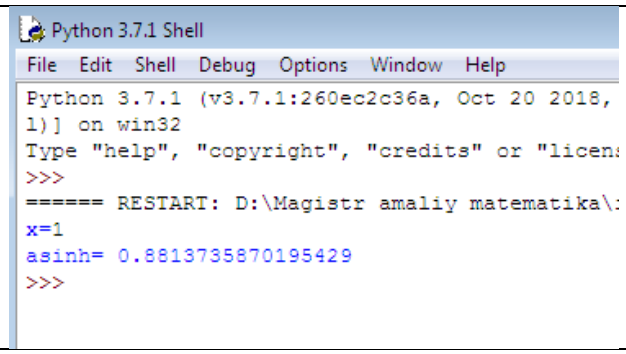
- asin() - radianda ifodalangan arksinus.

```
import math
x=int(input('x='))
y=math.asin(x)
print('asin=',y)
```



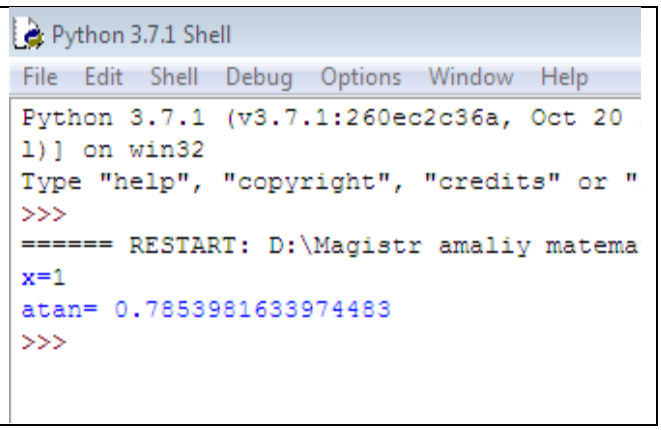
- asinh() - giperbolik arksinus.

```
import math
x=int(input('x='))
y=math.asinh(x)
print('asinh=',y)
```



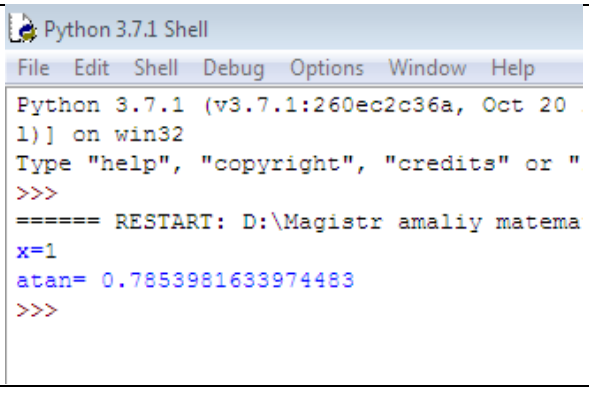
- atan() - radianda ifodalangan arktangis.

```
import math
x=int(input('x='))
y=math.atan(x)
print('atan=',y)
```



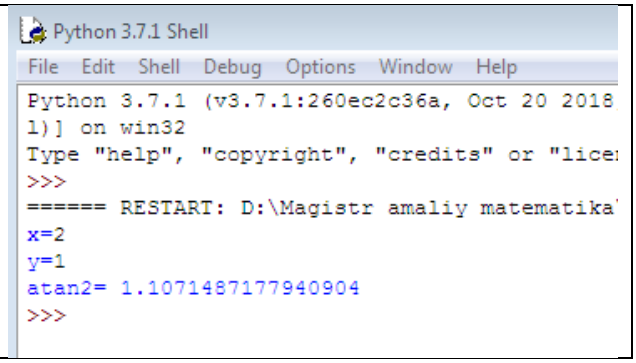
- atanh() - giperbolik arktanges.

```
import math
x=float(input('x='))
y=math.atanh(x)
print('atanh=',y)
```



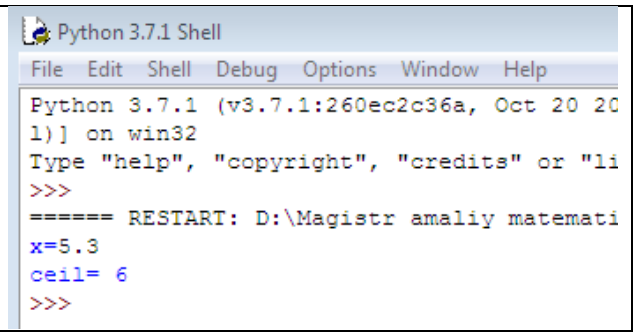
- atan2() - arktangens y/x ni, y va x kvadratlar ishorasi bilan aniqlanuvchi natijaviy kvadrat bilan qaytariladi.

```
import math
x=int(input('x='))
y=int(input('y='))
atan2=math.atan2(x,y)
print('atan2=',atan2)
```



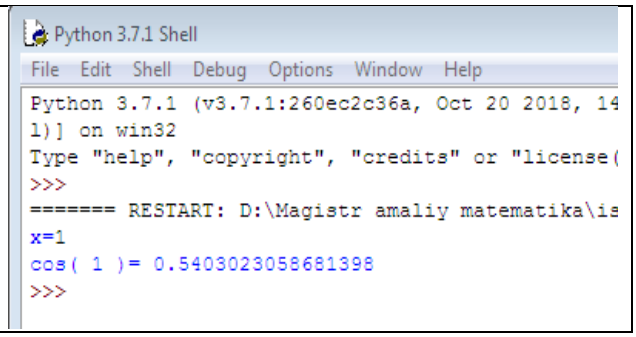
- ceil() - sonni o‘zidan katta butun songa yaxlitlash.

```
import math
x=float(input('x='))
ceil=math.ceil(x)
print('ceil=',ceil)
```



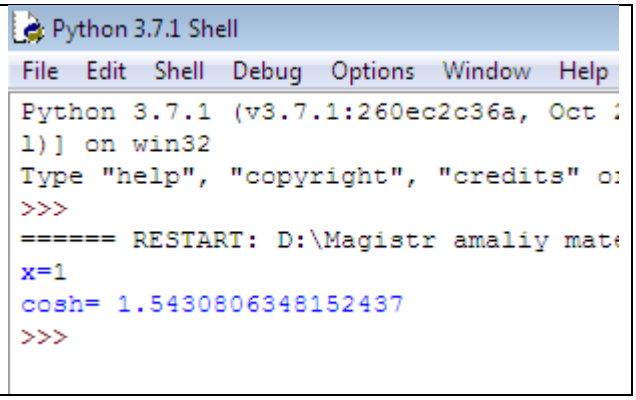
- cos() - radianda ifodalangan kosinus.

```
import math
x=int(input('x='))
y=math.cos(x)
print('cos(',x,')=',y)
```



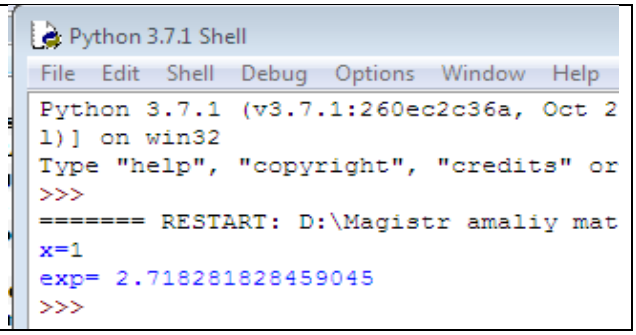
- cosh() - giperbolik kosinus.

```
import math
x=int(input('x='))
y=math.cosh(x)
print('cosh=',y)
```



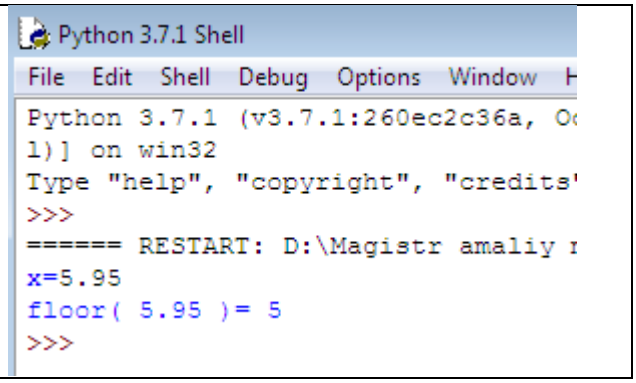
- exp() - berilgan sonning eksponentasini hisoblash.

```
import math
x=int(input('x='))
y=math.exp(x)
print('exp=',y)
```



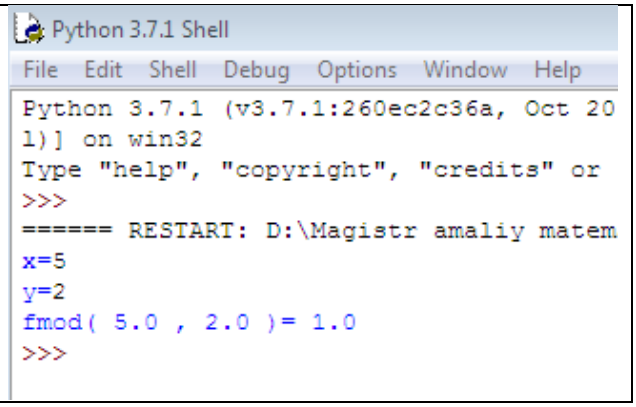
- floor() - sonni o‘zidan kichik butun songa yaxlitlash.

```
import math
x=float(input('x='))
y=math.floor(x)
print('floor(',x,')=',y)
```



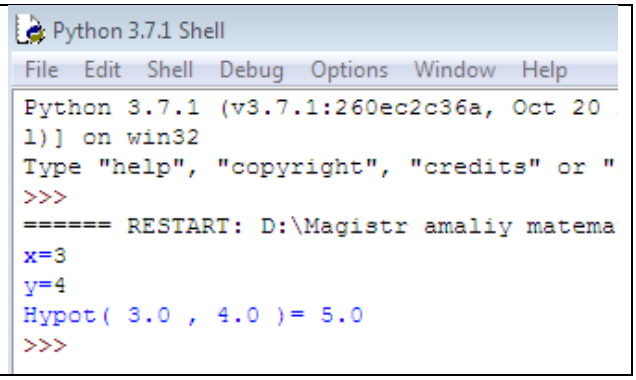
- fmod() -ikki son x ni y ga bo‘lgandagi qoldiqni hisoblaydi.

```
import math
x=float(input('x='))
y=float(input('y='))
natija=math.fmod(x,y)
print('fmod(',x,',',y,')=',natija)
```



- hypot() - to‘g‘ri burchakli uchburchakda ikki katet bo‘yicha gipotenuzani hisoblash.

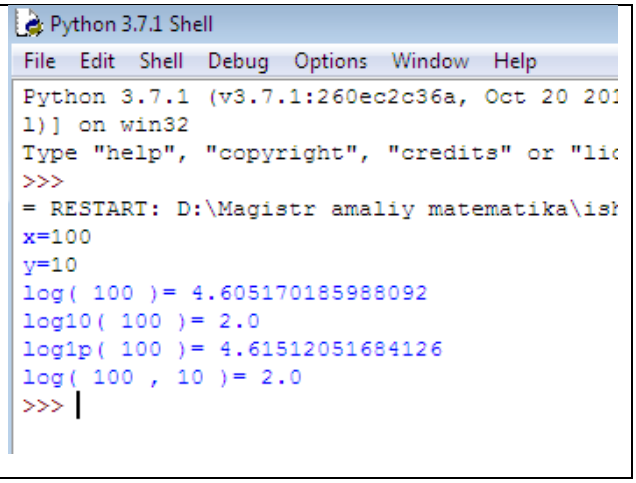
```
import math
x=float(input('x='))
y=float(input('y='))
natija=math.hypot(x,y)
print('Hypot(',x,',',y,')=',natija)
```



- log10() - o‘nlik logarifm.

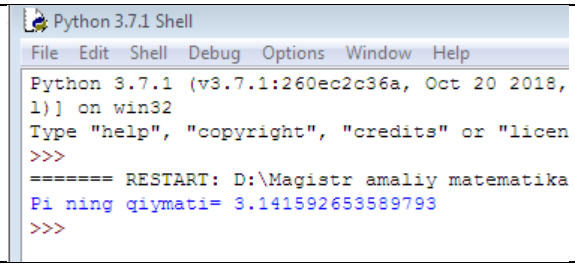
- log() - natural logarifm.
- log1p() – log(1+x), bunda x ning qiymati nolga yaqin bo‘lganda ham natija aniq chiqadi. log() ning aniqligi etarli bo‘lmaganligi sababli, bu holda shunchaki log(1) ga qaytiladi.

```
import math
x=int(input('x='))
y=int(input('y='))
y1=math.log(x)
y2=math.log10(x)
y3=math.log1p(x)
y4=math.log(x,y)
print('log(',x,')=',y1)
print('log10(',x,')=',y2)
print('log1p(',x,')=',y3)
print('log(',x,',',y,')=',y4)
```



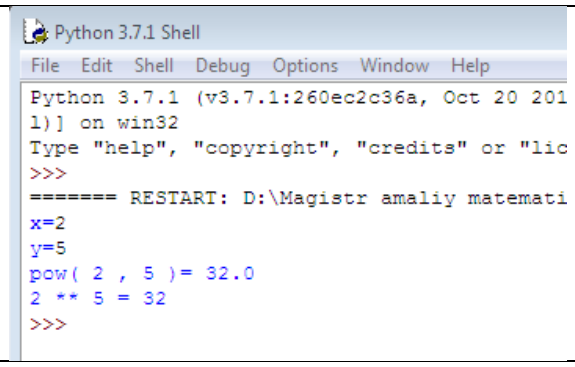
- pi() - π sonining qiymatini aniqlaydi.

```
import math
y=math.pi
print('Pi ning qiymati=',y)
```



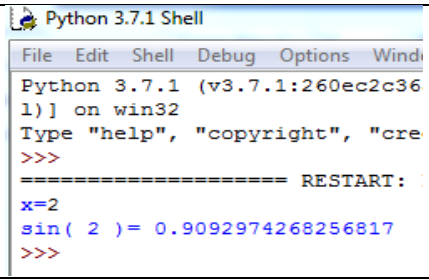
- pow() – x sonini y darajaga ko‘tarish.

```
import math
x=int(input('x='))
y=int(input('y='))
natija=math.pow(x,y)
print("pow(",x,",",y,")=",natija)
print(x,'**',y,'=',x**y)
```



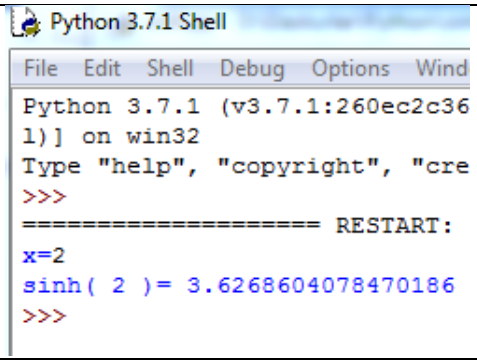
- sin() - radianda ifodalangan sinus.

```
import math
x=int(input('x='))
y=math.sin(x)
print('sin(',x,')=',y)
```



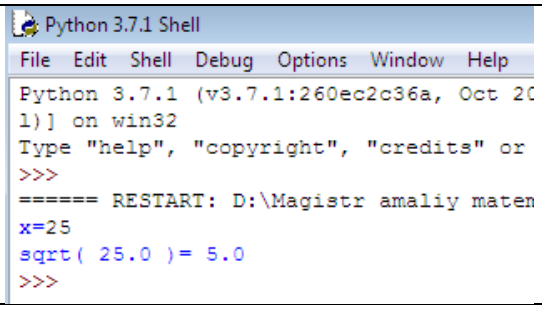
- sinh() - radianda ifodalangan geperbolik sinus.

```
import math
x=int(input('x='))
y=math.sinh(x)
print('sinh(',x,')=',y)
```



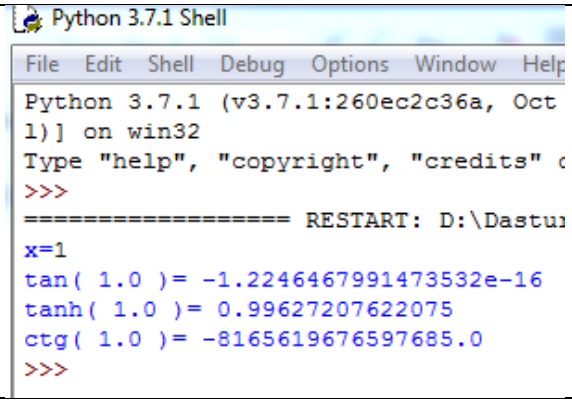
- sqrt() – x sonining kvadrat ildizi.

```
import math
x=float(input('x='))
y=math.sqrt(x)
print('sqrt(',x,')=',y)
```



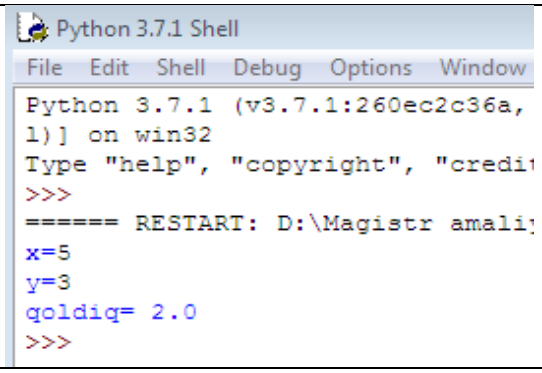
- tan() - radianda ifodalangan tangens
- tanh() - radianda ifodalangan giperbolik tangens.

```
import math
x=float(input('x='))
tan=math.tan(math.pi/x)
tanh=math.tanh(math.pi/x)
print('tan(',x,')=',tan)
print('tanh(',x,')=',tanh)
print('ctg(',x,')=',1/tan)
```



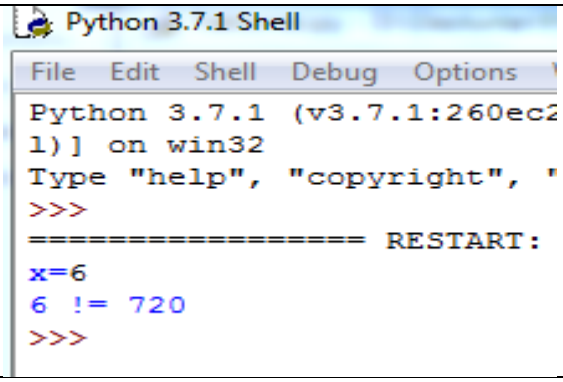
- % - birinchi argumentni ikkinchi argumentga bo‘lgandagi qoldiq.

```
import math
x=float(input('x='))
y=float(input('y='))
qoldiq=x % y
print('qoldiq=',qoldiq)
```



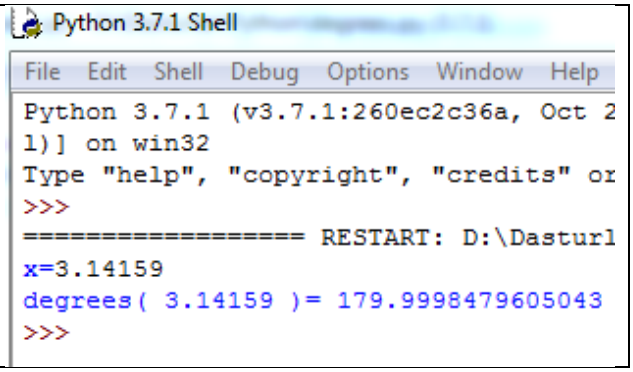
- factorial(num) – sonning faktorialini hisoblaydi.

```
import math
x=int(input('x='))
y=math.factorial(x)
print(x,'!=',y)
```



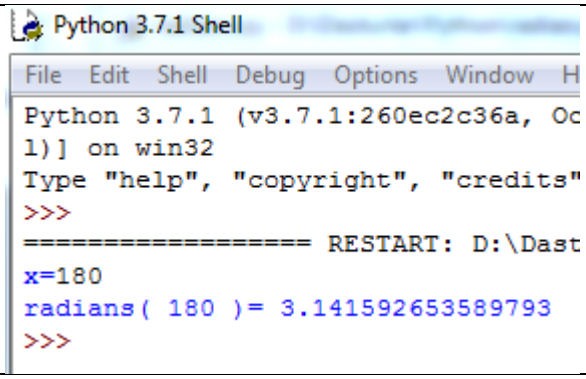
- degrees(rad) – radiandan gradusga o‘tkazadi.

```
import math
x=float(input('x='))
y=math.degrees(x)
print('degrees(',x,')=',y)
```



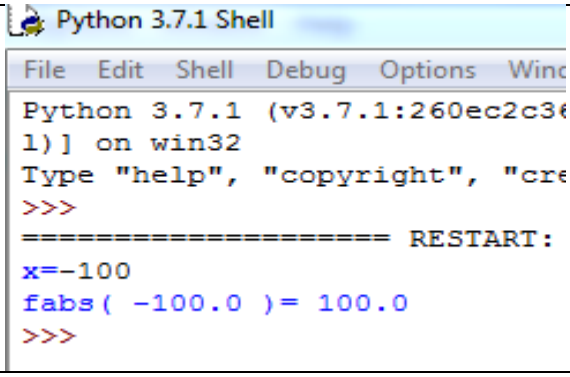
- radians(grad) – gradusdan radianga o‘tkazadi;

```
import math
x=int(input('x='))
y=math.radians(x)
print('radians(',x,')=',y)
```



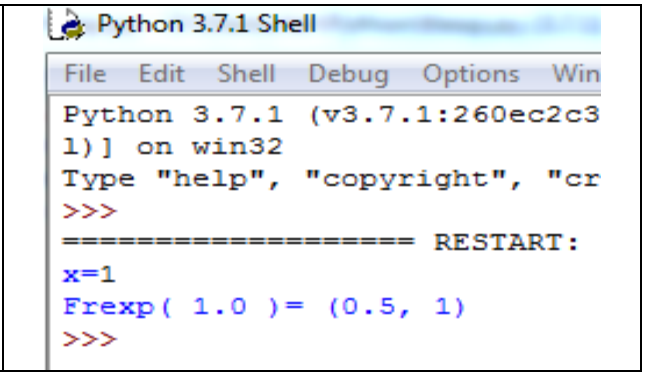
- Fabs(x) – x ning absolyut raqami

```
import math
x=float(input('x='))
y=math.fabs(x)
print('fabs(',x,')=',y)
```



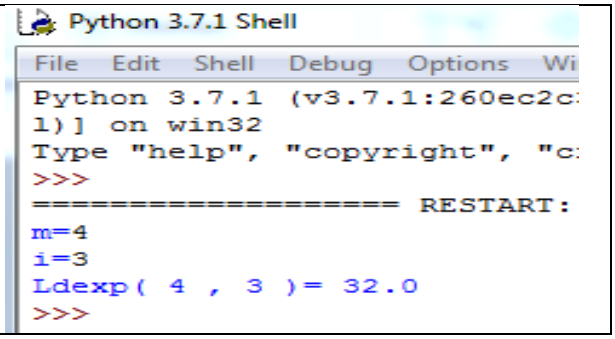
- Frexp(x) - mantisa va tartibni (m, i) juftligi kabi qaytaradi, m- o‘zgaruvchan nuqtali son, i esa- x=m*2**i ga teng butun son bo‘ladi. Agarda 0-(0,0) qaytarsa boshqa paytda 0.5<=abs(m)<1.0 bo‘ladi.

```
import math
x=float(input('x='))
frexp=math.frexp(x)
print('Frexp(',x,')=',frexp)
```



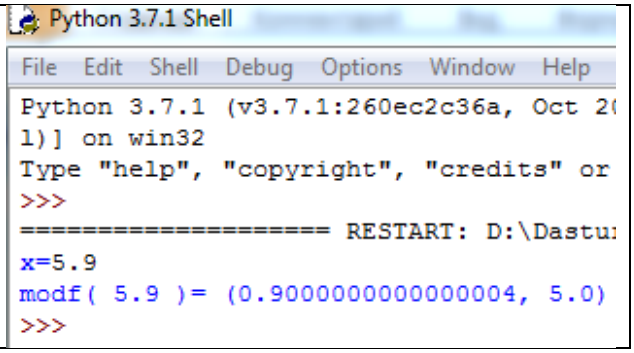
- Ldexp(m,i)=m*(2**i).- m ni, 2 ni i daragacha tartibda qaytaradi.

```
import math
m=int(input('m='))
i=int(input('i='))
Ldexp=math.ldexp(m,i)
print('Ldexp(',m,',',i,')=',Ldexp)
```



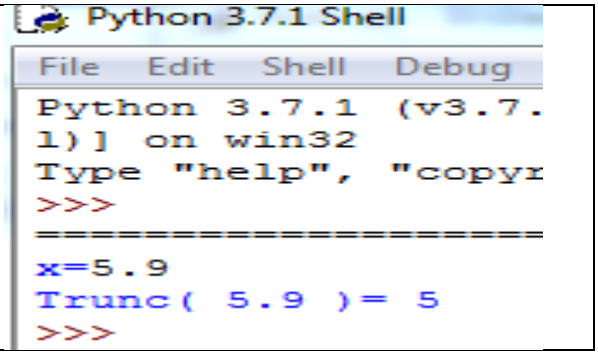
- Modf(x)- (y,q) juftlikda x ning butun va q kasr qismini qaytaradi.

```
import math
x=float(input('x='))
natija=math.modf(x)
print('modf(',x,')=',natija)
```



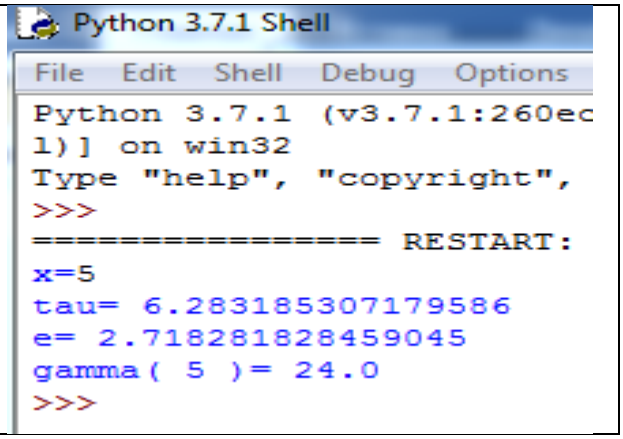
- Trunc(x)-x haqiqiy sonning butun qismini qaytaradi.

```
import math
x=float(input('x='))
y=math.trunc(x)
print('Trunc(',x,')=',y)
```



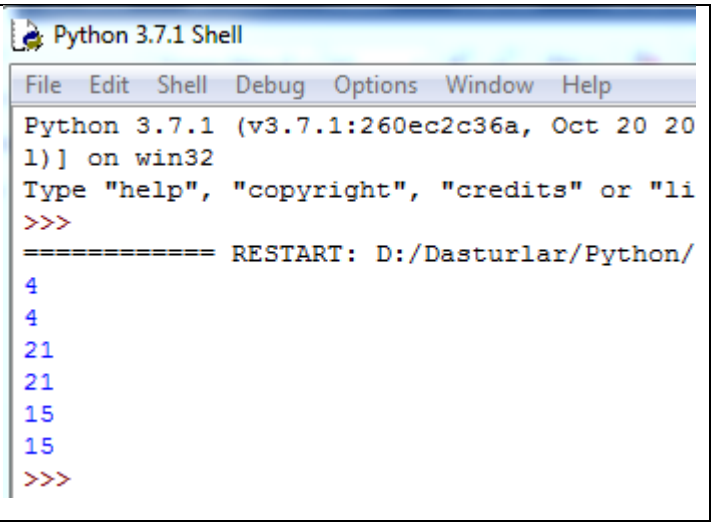
- tau – tau konstantasi
- e – e konstantasi
- gamma – x sonining gamma qiymati

```
import math
x=int(input('x='))
tau=math.tau
e=math.e
gamma=math.gamma(x)
print('tau=',tau)
print('e=',e)
print('gamma(',x,')=',gamma)
```



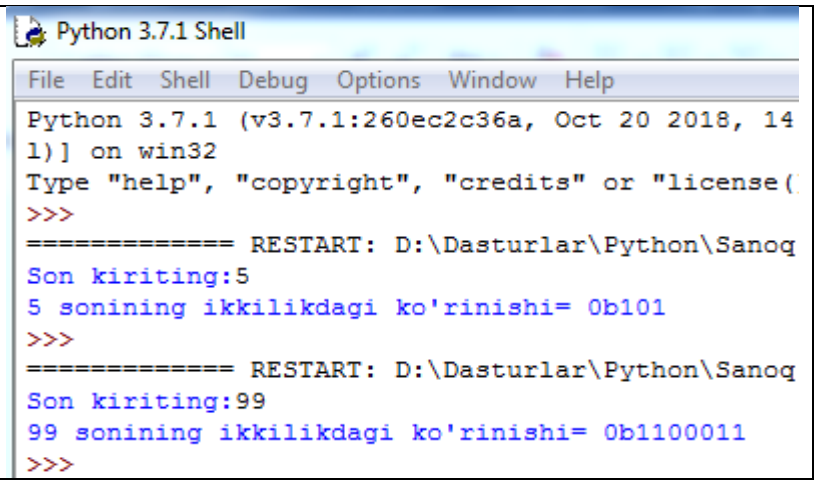
- int([object],[sanoq sistemasi asosi])- *butun sonni berilgan sanoq sistemasidan o‘nlik sanoq sistemasiga o‘tkazadi.*

```
ikkilik1=int('100',2);
print(ikkilik1);
ikkilik2=int('0b100',2);
print(ikkilik2);
sakkizlik1=int('25',8);
print(sakkizlik1);
sakkizlik2=int('0o25',8);
print(sakkizlik2);
un_olti1=int('F',16);
print(un_olti1);
un_olti2=int('0xF',16);
print(un_olti2);
```



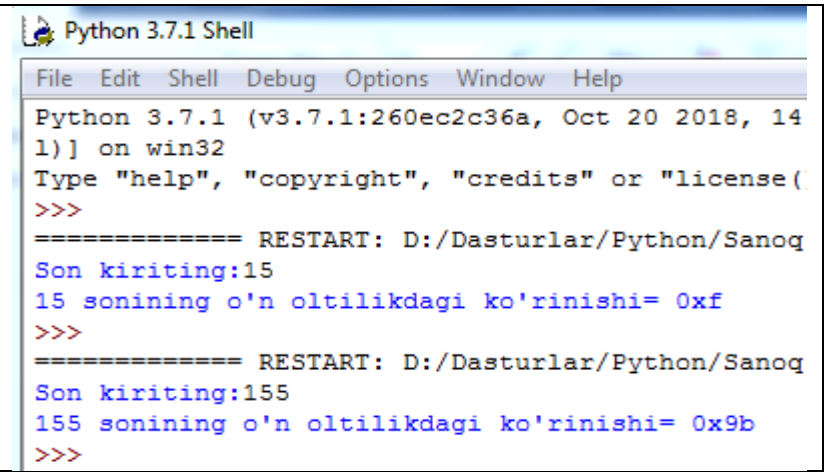
- bin(x)- butun sonni ikkilik sanoq sistemasiga o‘tkazadi.

```
n=int(input('Son kiriting:'));
s=bin(n);
print(n,"sonining ikkilikdagi
ko'rinishi=",s);
```



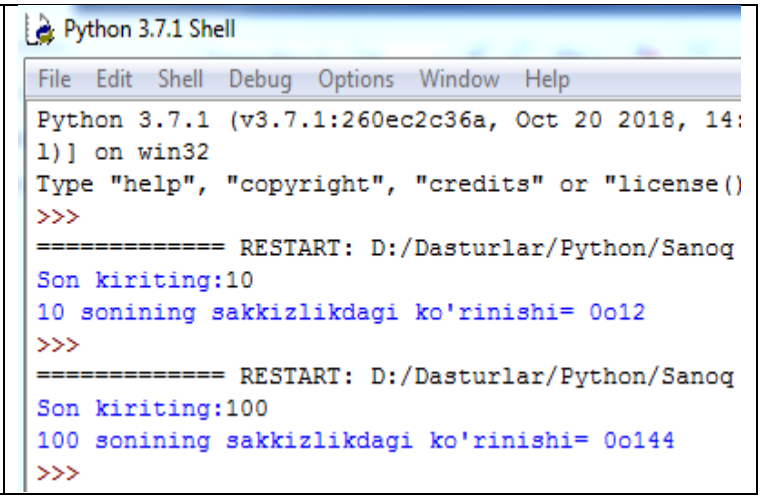
- hex(x)- butun sonni o‘n oltilik sanoq sistemasiga o‘tkazadi.

```
n=int(input('Son
kiriting:'));
s=hex(n);
print(n,"sonining o'n
oltilikdagi ko'rinishi=",s);
```



- oct(x)- butun sonni sakkizlik sanoq sistemasiga o‘tkazadi.

```
n=int(input('Son kiriting:'));
s=oct(n);
print(n,"sonining sakkizlikdagi
ko'rinishi=",s);
```



PYTHON tilida turli tipdagi o‘zgaruvchilardan foydalanish mumkin, shu sababli, har bir tipdagi o‘zgaruvchilar qanday tavsiflanishini bilish zarur.

PYTHON tilida bitta o‘zgaruvchini dastur bajarilishi davomida satr yoki son sifatida ishlatish mumkin. Shu bilan birga **PYTHON** tilida o‘zgaruvchilar bilan ishlanganda oshkor ko‘rsatilishi mumkin bo‘lgan asosiy ma’lumotlar tiplari to‘plami mavjud.

- **Butun (integer) sonlar** – Sonning kasr bo‘lmagan son bo‘lib, ularda sonning asosi (10 lik), o‘n oltilik (asosi 16-prefiksga ega) yoki sakkizlik (asosi 8-prefiksli) sanoq sistemalar ko‘rsatiladi.
- **Siljuvchi vergulli (float) sonlar** – sonning kompyuterda amalga oshiriladigan eksponentsiol yozuvi. Xuddi shuningdek, “ikkilangan aniqlikga” ega bo‘lgan son ham mavjud.
- **Satr (string)** – simvollar ketma – ketligidan iborat bo‘lib, unda har bir simvol bir bayt o‘lchamdan, toki maksimal uzunlik 216 gacha bo‘lgan joyni egallaydi. Yakka qavslarga olingan satrlar literallar sifatida qaraladi, ayni paytda

ikkilangan qavslar ichidagi satrlar esa (maxsus belgilar, o‘zgaruvchilarning qiymatlari va shu kabilar) sifatida talqin qilinadi.

- **Bul (boolean) tipi -** Mantiqiy ifoda bo‘lib, uning qiymati faqat rost (True) yoki yolg‘on (False) dan iborat.
- **Kompleks (complex) tipi** – Sonning birinchi argument sifatida haqiqiy qism, ikkinchi argument sifatida mavhum qism uzatiladi.
- **Massiv (array)** – bir nechta qiymatlarning tartiblashtirilgan xaritasi bo‘lib, undagi kalitlar qiymatlarga mos keladi. Kalitlar – bu indeks nomerlari (ular so‘zsiz tushuniladi) yoki aniq ko‘rsatilgan nishonlar.
  misol
- **Ob’ekt –** bu berilganlarning xossalarini saqlovchi va berilganlarni qayta ishlash metodlaridan iborat bo‘lgan sinf.
- **Resurs –** tashqi resursga havola bo‘lib, ular maxsus funksiyalar tomonidan yaratiladi va saqlanadi.
- **NULL –** qiymatga ega bo‘lmagan o‘zgaruvchi. Bu o‘zgaruvchi, shakllantirilmagan bo‘ladi (unga hech bir qiymat berilmagan bo‘ladi), agar unga NULL o‘zgarmasi ta’minlangan bo‘lsa yoki unset() funksiyasi yordamida bekor qilinmagan bo‘lsa.

2.2. PYTHON DA ARIFMETIK, MANTIQIY OPERATORLAR VA ULARNING TADBIQI

Arifmetik amallar va qiymat berish operatori. Berilganlarni qayta ishlash uchun PYTHON tilida amallarning juda keng majmuasi aniqlangan. Amal - bu qandaydir harakat bo‘lib, u bitta (unar) yoki ikkita (binar) operandlar ustida bajariladi, hisob natijasi uning qaytariluvchi qiymati hisoblanadi. Tayanch arifmetik amallarga qo‘shish (+), ayirish (-), ko‘paytirish (*), bo‘lish (/), darajaga ko‘tarish (**) va bo‘lish qoldig‘ini olish (%) amallarini keltirish mumkin. Amallar qaytaradigan qiymatlarni o‘zlashtirish uchun qiymat berish amali (=) va uning turli modifikatsiyalari ishlatiladi: qo‘shish, qiymat berish bilan (+); ayirish, qiymat berish bilan (-); ko‘paytirish qiymat berish bilan (*); bo‘lish, qiymat berish bilan (/); bo‘lish qoldig‘ini olish, qiymat berish bilan (%) va boshqalar. Ularning umumiy ko‘rinishlariga to‘xtalamiz.

Razryadli mantiqiy amallar. Dastur tuzish tajribasi shuni ko‘rsatadiki, odatda qo‘yilgan masalani yechishda biror holat ro‘y berganligini yoki yo‘qligini ifodalash uchun 0 va 1 qiymat qabul qiluvchi bayroqlardan foydalaniladi. Shu maqsadda bir yoki undan ortiq baytli o‘zgaruvchilardan foydalanish mumkin. Masalan, bool (mantiqiy) tupdagi o‘zgaruvchini shu maqsadda ishlatsa bo‘ladi. Boshqa tomondan, bayroq sifatida baytning razryadlaridan foydalanish ham mumkin. Chunki razryadlar faqat ikkita qiymatni – 0 va 1 sonlarini qabul qiladi. Bir baytda 8 razryad bo‘lgani uchun unda 8 ta bayroqni kodlash imkoniyati

mavjud. Quyidagi jadvalda Python tilida bayt razryadlari ustida mantiqiy amallar majmuasi keltirilgan.

Bayt razryadlari ustida mantiqiy amallar

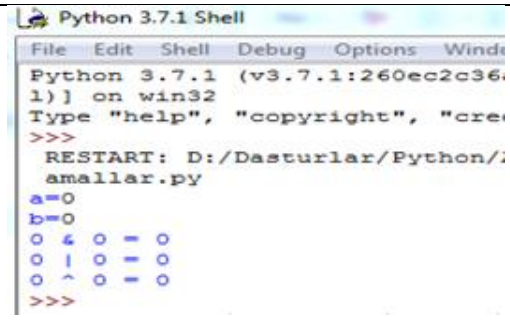
Amallar	Mazmuni
And yoki **&**	Mantiqiy VA (ko'paytirish)
Xor yoki **|**	Mantiqiy yoki (qo'shish))
Or yoki **^**	Istisno qiluvchi YOKI

Razryadli mantiqiy amallarning bajarish natijalarini jadval ko'rinishida ko'rsatish mumkin.

A	**B**	**A&B**	**A|B**	**A^B**
0	**0**	**0**	**0**	**0**
0	**1**	**0**	**1**	**1**
1	**0**	**0**	**1**	**1**
1	**1**	**1**	**1**	**0**

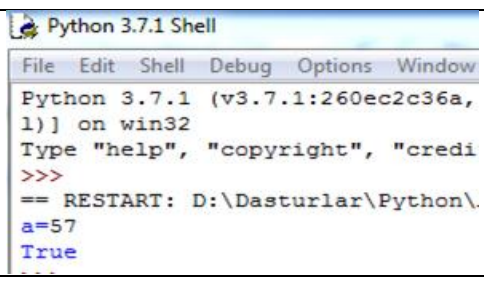
Razryadli mantiqiy amallarni bajarish natijalari

```
a=int(input('a='))
b=int(input('b='))
print(a,'&',b,'=',a&b)
print(a,'|',b,'=',a|b)
print(a,'^',b,'=',a^b)
```



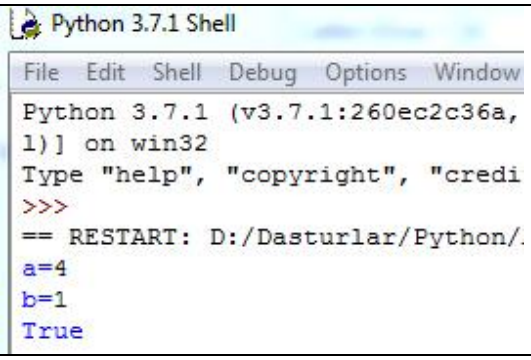
1.1-masala. A butun soni berilgan. Jumlani rostlikka tekshiring: "A soni toq son".

```
a=int(input('a='))
print(bool(a%2==1))
```



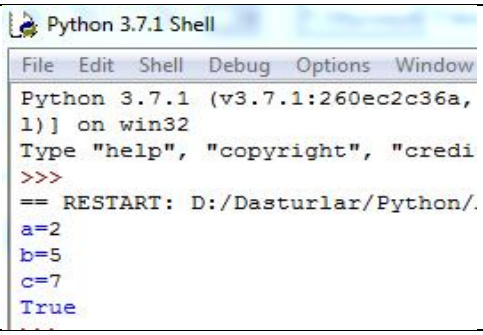
1.2-masala. Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshiring: “A>2 va B<=3”.

```
a=int(input('a='))
b=int(input('b='))
c=bool(a>2 and b<=3)
print(c)
```



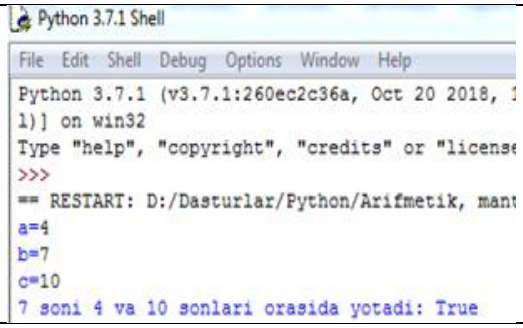
1.3-masala Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshiring: “A<=B<=C”

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
hisoblash=bool(a<=b and b<=c)
print(hisoblash)
```



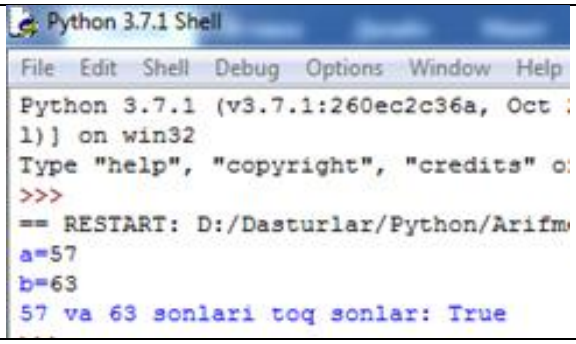
1.4-masala. Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshiring: “B soni A va C sonlari orasida yotadi”.

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
natija=bool(a<b and b<c)
print(b,'soni',a,'va',c,'sonlari orasida
yotadi:',natija)
```



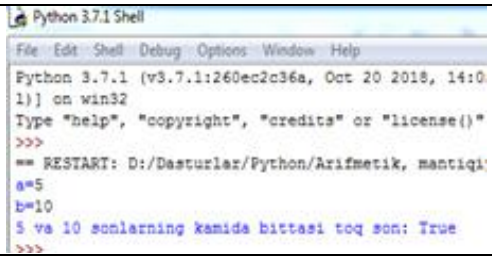
1.5-masala. Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshiring: “A va B sonlari toq sonlar”.

```
a=int(input('a='))
b=int(input('b='))
toq_son=bool(a%2==1 and b%2==1)
print(a,'va',b,'sonlari toq
sonlar:',toq_son)
```



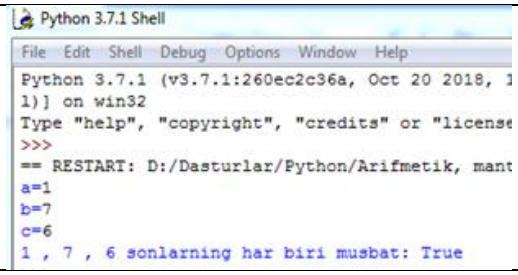
1.6-masala Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshiring: “A va B sonlarning kamida bittasi toq son”.

```
a=int(input('a='))
b=int(input('b='))
bitta_toq=bool(a%2==1 or b%2==1)
print(a,'va',b,'sonlarning kamida bittasi
toq son:',bitta_toq)
```



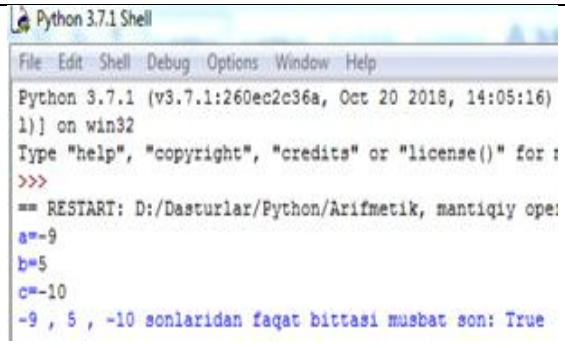
1.7-masala Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshiring: “A, B, C sonlarning har biri musbat”.

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
musbat=bool(a>0 and b>0 and c>0)
print(a,',',b,',',c,'sonlarning har biri
musbat:',musbat)
```



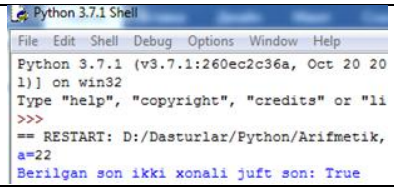
1.8-masala Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshiring: “A, B, C sonlaridan faqat bittasi musbat son”.

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
bitta_musbat=bool((a>0 and b<0 and
c<0)or(a<0 and b>0 and c<0)or(a<0 and
b<0 and c>0))
print(a,',',b,',',c,'sonlaridan faqat bittasi
musbat son:',bitta_musbat)
```



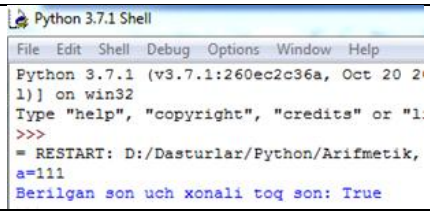
1.9-masala. Musbat butun son berilgan. Jumlani rostlikka tekshiring: “Berilgan son ikki xonali juft son”.

```
a=int(input('a='))
b=bool(a>9 and a<100 and a%2==0)
print('Berilgan son ikki xonali juft son:',b)
```



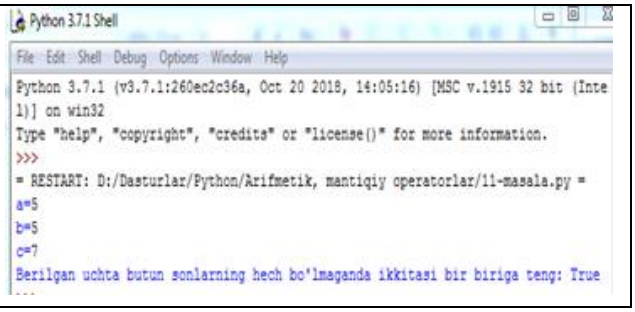
1.10-masala. Musbat butun son berilgan. Jumlani rostlikka tekshiring: “Berilgan son uch xonali toq son”.

```
a=int(input('a='))
b=bool(a>99 and a<1000 and a%2==1)
print('Berilgan son uch xonali toq son:',b)
```



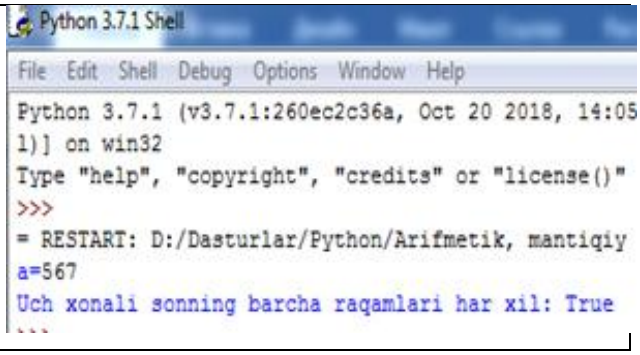
1.11-masala. Jumlani rostlikka tekshiring: “Berilgan uchta butun sonlarning hech bo‘lmaganda 2 tasi bir biriga teng”.

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
teng=bool(a==b or a==c or b==c)
print("Berilgan uchta butun sonlarning
hech bo'lmaganda ikkitasi bir biriga
teng:",teng)
```



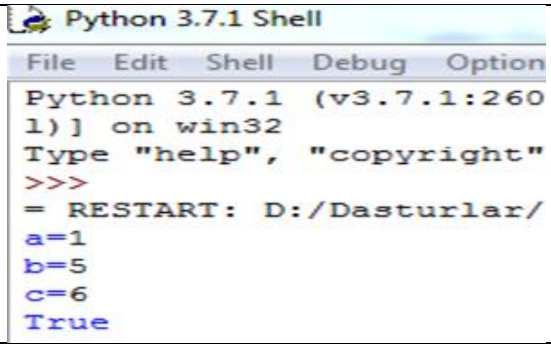
1.12-masala. Uch xonali son berilgan. Jumlani rostlikka tekshiring: “Ushbu sonning barcha raqamlari xar xil”.

```
import math
a=int(input('a='))
x=math.floor(a/100)
y=math.floor(a/10)%10
z=math.floor(a/10)
natija=bool(x!=y and x!=z and y!=z)
print('Uch xonali sonning barcha
raqamlari har xil:',natija)
```



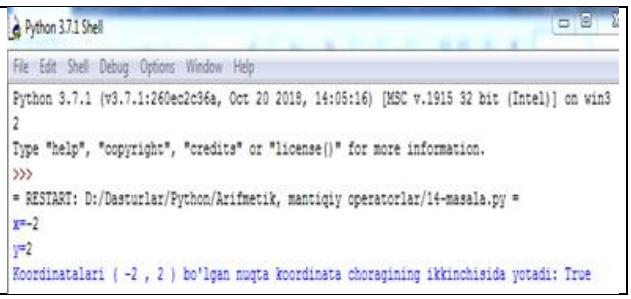
1.13-masala. A, B, C sonlar berilgan (A soni noldan farqli). $D=B^2-4AC$ diskriminantdan foydalanib, jumlani rostlikka tekshiring: “$Ax^2+Bx+C=0$ kvadrat tenglama haqiqiy ildizga ega”.

```
import math
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
d=math.pow(b,2)-4*a*c
natija=bool(a!=0 and d>=0)
print(natija)
```



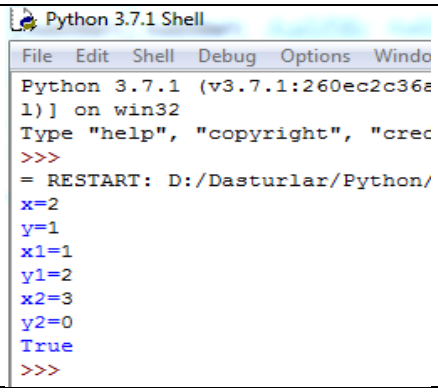
1.14-masala. x, y sonlar berilgan. Jumlani rostlikka tekshiring: “Koordinatalari (x,y) bo‘lgan nuqta, koordinata choragining ikkinchisida yotadi”.

```
x=int(input('x='))
y=int(input('y='))
chorak_2=bool(x<0 and y>0)
print('Koordinatalari (',x,',',y,") bo'lgan
nuqta koordinata choragining
ikkinchisida yotadi:",chorak_2)
```



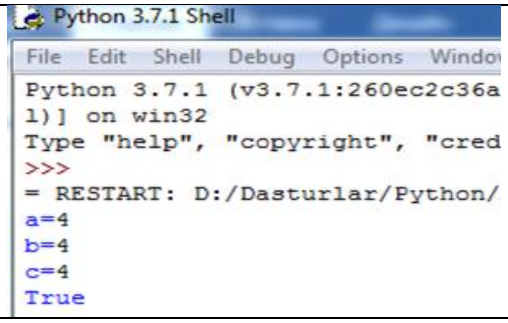
1.15-masala. (x, y), (x1, y1), (x2, y2) sonlari berilgan. Jumlani rostlikka tekshiring: “Koordinatalari (x,y) bo‘lgan nuqta, chap yuqori cho’qqisi (x1,y1) koordinatalarga ega bo‘lgan va o‘ng pastikisi (x2,y2) bo‘lgan, tomonlari esa koordinata o‘qlariga parallel bo’lgan to‘rtburchak ichida yotadi”.

```
x=int(input('x='))
y=int(input('y='))
x1=int(input('x1='))
y1=int(input('y1='))
x2=int(input('x2='))
y2=int(input('y2='))
print(bool(x>x1 and x2>x and y1>y and y>y2))
```



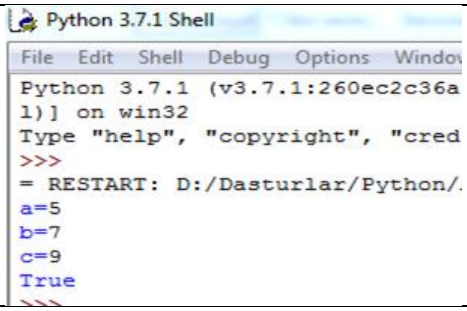
1.16-masala. a, b, c butun sonlari berilgan. Jumlani rostlikka tekshiring: “a, b, c tomonli uchburchak teng tomonli bo‘ladi”.

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
print(bool(a==b and a==c and b==c))
```



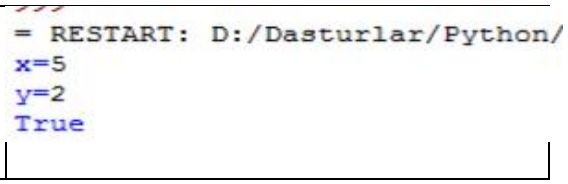
1.17-masala. a, b, c butun sonlar berilgan. Jumlani rostlikka tekshiring: “a, b, c tomonli uchburchak yasash mumkin”.

```
a=int(input('a='))
b=int(input('b='))
c=int(input('c='))
print(bool((a+b)>c or (a+c)>b or (b+c)>a))
```



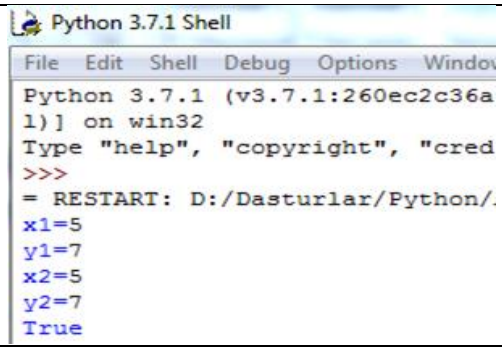
1.18-masala. Shaxmat doskasining x, y koordinatalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Doskaning chap pastki maydoni (1,1) qoraligini hisobga olib, jumlani rostlikka tekshiring: “Berilgan (x, y) koordinatali maydon oq”.

```
x=int(input('x='))
y=int(input('y='))
print(bool((x>=1 and x<=8 and y>=1 and
y<=8)and(x+y)%2==1))
```



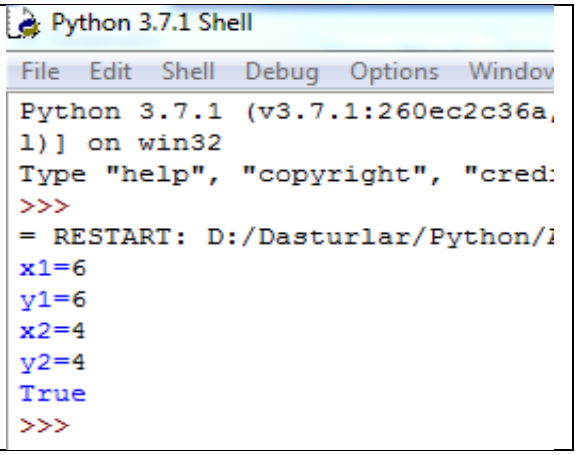
1.19-masala. Shaxmat doskasining ikkita turli (x1, y1), (x2, y2) koordinalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Jumlani rostlikka tekshiring: “Ruh bir yurishda bir maydondan ikkinchisiga o‘ta oladi”.

```
x1=int(input('x1='))
y1=int(input('y1='))
x2=int(input('x2='))
y2=int(input('y2='))
print(bool((x1>=1 and x1<=8 and x2>=1 and
x2<=8 and y1>=1 and y1<=8 and y2>=1 and
y2<=8) and (x1==x2 or y1==y2)))
```



1.20-masala. Shaxmat doskasining ikkita turli (x1, y1), (x2, y2) koordinalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Jumlani rostlikka tekshiring: “Ot bir yurishda bir maydondan ikkinchisiga o‘ta oladi”.

```
x1=int(input('x1='))
y1=int(input('y1='))
x2=int(input('x2='))
y2=int(input('y2='))
print(bool((x1>=1 and x1<=8 and y1>=1
and y1<=8 and x2>=1 and x2<=8 and
y2>=1 and y2<=8) and (abs(y2-y1)==2 or
abs(x2-x1)==2 and abs(y2-y1)==1)))
```



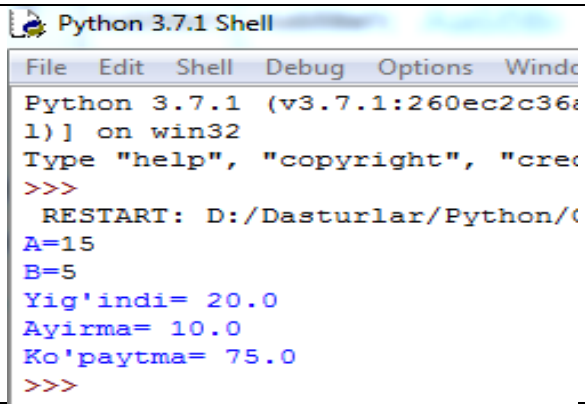
2.3. PYTHON DA CHIZIQLI ALGORITMLAR BILAN ISHLASH

Odatda tabiat yoki jamiyatda uchraydigan turli muammo, masala yoki jarayonlarni o‘rganishni kompyuter yordamida olib borish uchun, birinchi navbatda, qaralayotgan masala, jarayon - obyektning matematik ifodasi, ya’ni matematik modelini ko‘rish kerak bo‘ladi. Qaralayotgan obyektning matematik modelini yaratish juda murakkab jarayon bo‘lib, o‘rganilayotgan obyektga bog‘liq ravishda turli soha mutaxassislarining ishtiroki talab etiladi. Umuman, biror masalani kompyuter yordamida yechishni quyidagi bosqichlarga ajratish mumkin**.** Qo‘yilgan chiziqli masalani kompyuterda yechish uchun, avval uning matematik modelini, keyin algoritmini va dasturini tuzish kerak bo‘ladi. Har qanday murakkab algoritmni ham uchta asosiy struktura yordamida tasvirlash mumkin. Bular ketma-ketlik, ayri va takrorlash strukturalaridir. Bu strukturalar asosida chiziqli, tarmoqlanuvchi va takrorlanuvchi hisoblash jarayonlarining algoritmlarini tuzish mumkin.

2.1-masala. A va B ikkita haqiqiy sonlar berilgan. Ularning yig‘indisi, ayirmasi va ko‘paytmasini hisoblang.

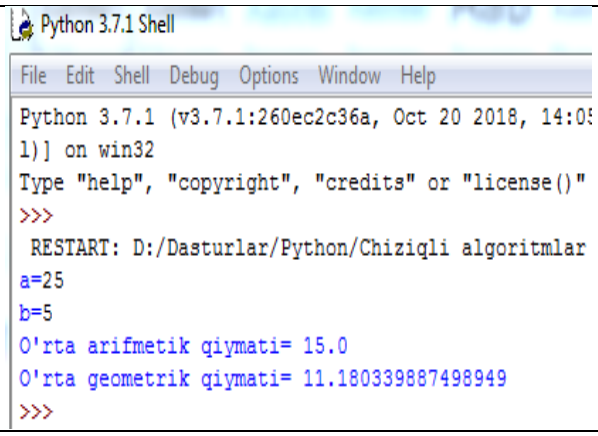
Yechish. a va b sonlar yig‘indisini S, ayirmasini d, ko‘paytmasini k bilan belgilasak, S=a+b, d=a-b, k=a*b formulalar o’rinli bo‘ladi.

```
a=float(input('A='))
b=float(input('B='))
s=a+b
d=a-b
k=a*b
print("Yig'indi=",s,'\nAyirma=',d,"\nKo'paytma=",k)
```



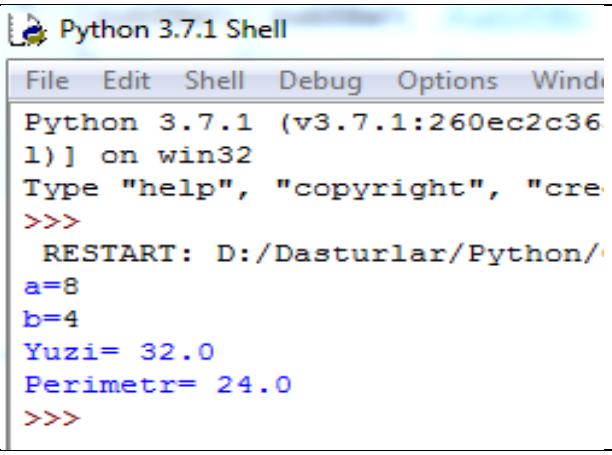
2.2-masala. Ikkita musbat son berilgan, bu sonlarning o‘rta arifmetik va o‘rta geometrik qiymatlarini aniqlang.
Yechish. a va b sonlarning o‘rta arifmetik qiymatini c, o‘rta geometrik qiymatini d bilan belgilasak, $c=\frac{a+b}{2}$; $d=\sqrt{a\cdot b}$; formulalar o‘rinli bo‘ladi.

```
import math
a=float(input('a='))
b=float(input('b='))
s=(a+b)/2
d=math.sqrt(a*b)
print("O'rta arifmetik qiymati=",s,"\nO'rta geometrik qiymati=",d)
```



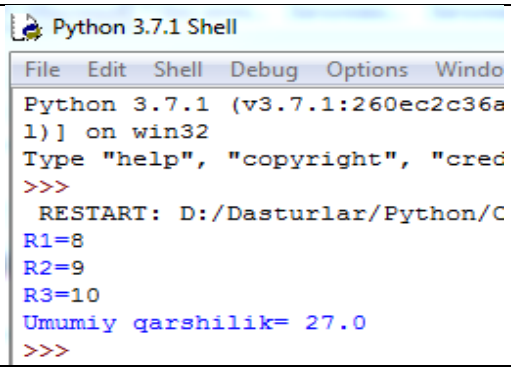
2.3-masala. Tomonlari A va B ga teng to‘g‘ri to‘rtburchakning yuzi va peremetri hisoblang. Yechish. To‘g‘ri to‘rtburchakning yuzi $s=a\cdot b$, peremetri $p=2\cdot(a+b)$ formulalar yordamida aniqlanadi.

```
a=float(input('a='))
b=float(input('b='))
s=a*b
p=2*(a+b)
print('Yuzi=',s,'\nPerimetr=',p)
```



2.4-masala. R1, R2, R3 uchta qarshiliklar ketma-ket ulangan zanjirning qarshiligini aniqlang. Yechish. Zanjirning umumiy qarshiligini R bilan belgilasak, ketma-ket ulashda $R = R_1 + R_2 + R_3$ formulalar o‘rinli bo‘ladi.

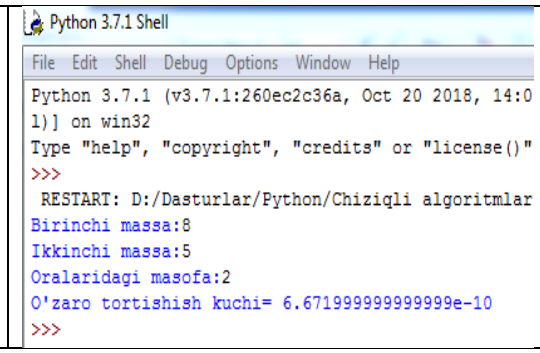
```
R1=float(input('R1='))
R2=float(input('R2='))
R3=float(input('R3='))
R=R1+R2+R3
print('Umumiy qarshilik=',R)
```



2.5-masala. Massalari M1 va M2 (kg) ga teng, oralaridagi masofa R (m) ga teng bo’lgan ikkita jismning o‘zaro tortishish kuchi F ni aniqlang. Bunda gravitatsion doimiysini G=6,672·10^{-11} (N·m^2/kg^2) deb oling.

Yechish. Butun olam tortilish qonuniga ko‘ra $F = G\frac{m_1 \cdot m_2}{R^2}$; yerning massasi $m_1 = 5{,}97 \cdot 10^{24}$, oyning massasi $m_2 = 7{,}35 \cdot 10^{22}$, yer bilan oy orasidagi masofa $R = 3{,}844 \cdot 10^8$. Izoh. Yer bilan Oyning massalari kilogrammda, masofa mertda, kuch Nyutonda o‘lchanadi.

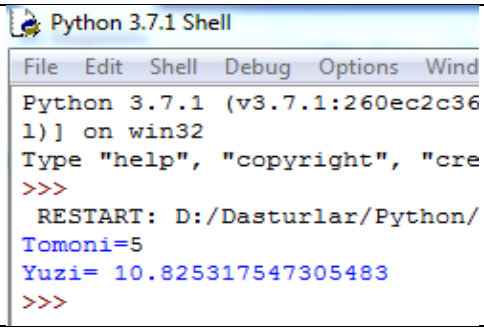
```
import math
M1=float(input('Birinchi massa:'))
M2=float(input('Ikkinchi massa:'))
R=float(input('Oralaridagi masofa:'))
G=6.672*math.pow(10,-11)
F=(G*M1*M2)/(math.pow(R,2))
print("O'zaro tortishish kuchi=",F)
```



2.6-masala. Teng tomonli uchburchakning tomoni A ga teng. Uchburchakning yuzini toping.

Yechish. Teng tomonli uchburchakning yuzini S bilan belgilasak, Formula o‘rinli bo‘ladi. $S = a^2 \cdot \frac{\sqrt{3}}{4}$

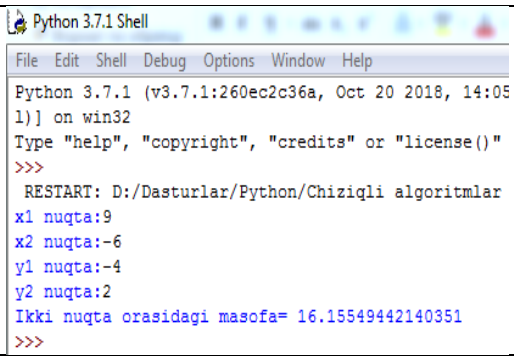
```
import math
a=float(input('Tomoni='))
S=math.pow(a,2)*math.sqrt(3)/4
print('Yuzi=',S)
```



2.7-masala. Koordinatalari X_1,Y_1 va X_2, Y_2 ga teng bo‘lgan nuqtalari orasidagi masofani hisoblang.

Yechish. Ikki nuqta orasidagi masofa $S=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$; formula yordamida aniqlanadi.

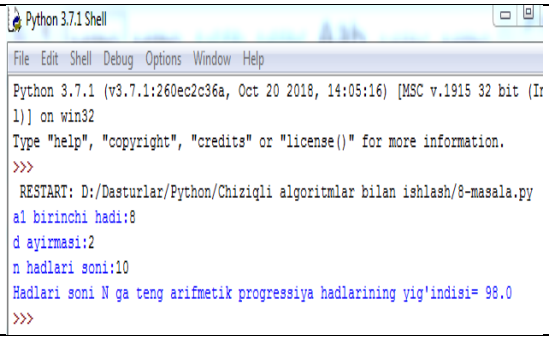
```
import math
x1=float(input('x1 nuqta:'))
x2=float(input('x2 nuqta:'))
y1=float(input('y1 nuqta:'))
y2=float(input('y2 nuqta:'))
s=math.sqrt(math.pow((x2-
x1),2)+math.pow((y2-y1),2));
print('Ikki nuqta orasidagi masofa=',s)
```



2.8-masala. Birinchi hadi A, ayirmasi D, hadlari soni N ga teng arifmetik progressiyaning hadlarining yig‘indisini hisoblang.

Yechish. Arifmetik progressiya istalgan hadi va hadlari yig‘indisi uchun $a_n=a+d\cdot(n-1)$, $S_n=\dfrac{2\cdot a+d\cdot(n-1)\cdot n}{2}$, formulalar o‘rinli bo‘ladi.

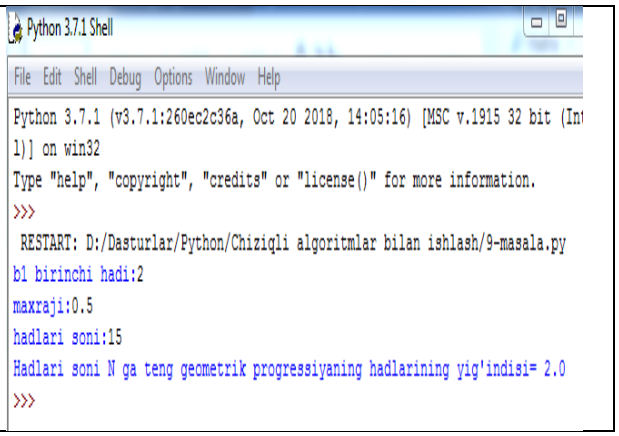
```
a1=float(input('a1 birinchi hadi:'))
d=float(input('d ayirmasi:'))
n=float(input('n hadlari soni:'))
s=(2*a1+d*(n-1)*n)/2
print("Hadlari soni N ga teng arifmetik
progressiya hadlarining yig'indisi=",s)
```



2.9-masala. Birinchi hadi B, maxraji Q va hadlari soni N ga teng geometrik progressiyaning hadlarining yig‘indisini hisoblang.

Yechish. Geometrik progressiyaning istalgan hadi va hadlari yig‘indisi $b_n=b\cdot q^{n-1}$; $s_n=\dfrac{b\cdot q-b}{q-1}$; formula yordamida aniqlanadi.

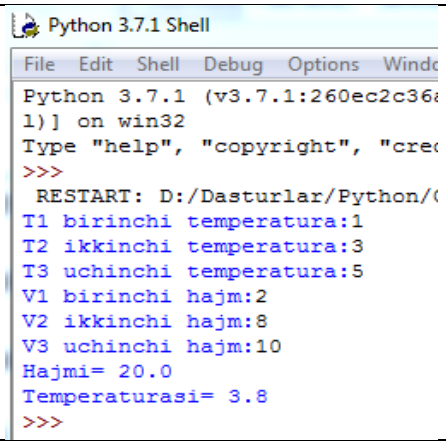
```
float(input('b1 birinchi hadi:'))
q=float(input('maxraji:'))
n=float(input('hadlari soni:'))
s=(b1*q-b1)/(q-1)
print("Hadlari soni N ga teng geometrik
progressiyaning hadlarining yig'indisi=",s)
```



2.10-masala. Uchta idishga suv solingan. Idishlardagi suvlarning temperaturasi mos ravishda T_1, T_2, T_3 ga, hajmi esa V_1, V_2, V_3 ga teng. Idishlardagi suvni bitta idishga quyilsa, uning hajmi va temperaturasi qanday bo‘ladi?

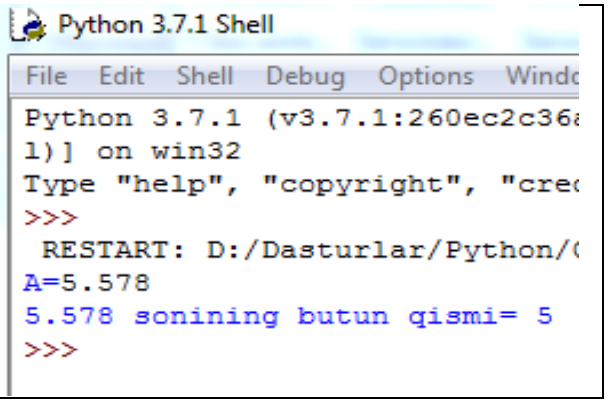
Yechish. Idishlardagi suvni bitta idishga quyilsa, suvning hajmi va temperaturasi $V = V_1 + V_2 + V_3;\ \ T = \frac{V_1 * T_1 + V_2 * T_2 + V_3 * T_3}{V}$ formulalar bilan aniqlanadi.

```
T1=float(input('T1 birinchi temperatura:'))
T2=float(input('T2 ikkinchi temperatura:'))
T3=float(input('T3 uchinchi temperatura:'))
V1=float(input('V1 birinchi hajm:'))
V2=float(input('V2 ikkinchi hajm:'))
V3=float(input('V3 uchinchi hajm:'))
V=V1+V2+V3
T=(V1*T1+V2*T2+V3*T3)/V
print('Hajmi=',V,'\nTemperaturasi=',T)
```



2.11-masala. Berilgan sonning butun qismini aniqlang. Yechish. A sonning butun qismini B bilan belgilasak, B=floor(A) formula bilan aniqlanadi.

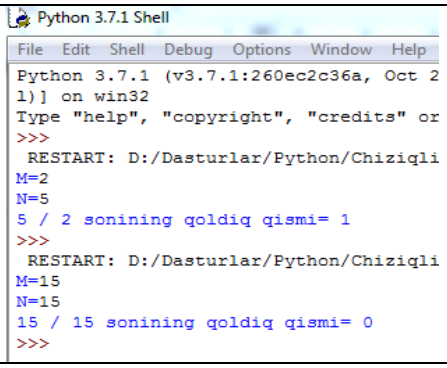
```
import math
A=float(input('A='))
B=math.floor(A)
print(A,"sonining butun qismi=",B)
```



2.12-masala. N/M ifodani hisoblashda hosil bo‘ladigan qoldiqni toping.

Yechish. Qoldiqni Z bilan belgilasak, u holda $Z = N - floor(\frac{N}{M}) \cdot M$ formula bilan hisoblanadi.

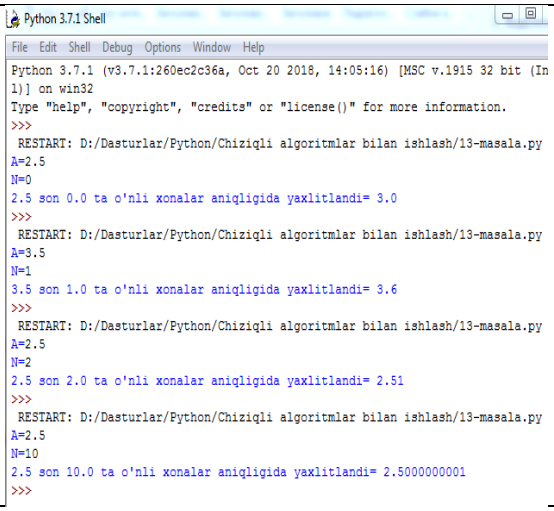
```
import math
M=int(input('M='))
N=int(input('N='))
Z=N-math.floor(N/M)*M
print(N,"/",M,"sonining qoldiq qismi=",Z)
```



2.13-masala. Sonni berilgan aniqlikda yaxlitlang.

Yechish. A sonni N ta o‘nli xonalar aniqligida yaxlitlash uchun $B=\frac{Ceil(A\cdot 10^N+0.5)}{10^N}$; formuladan foydalanamiz.

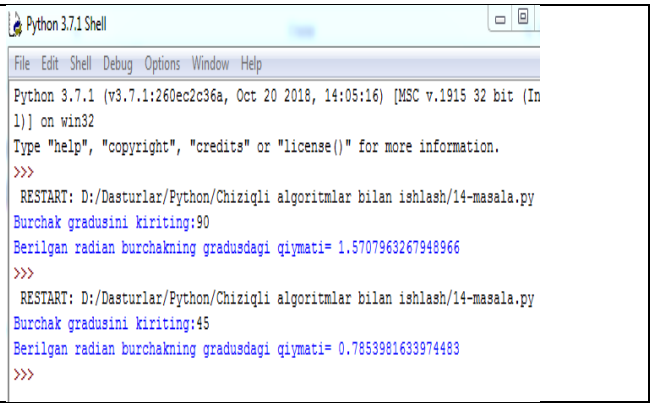
```
import math
A=float(input('A='))
N=float(input('N='))
B=math.ceil((A*math.pow(10,N)+0.5))/
math.pow(10,N)
print(A,"son",N,"ta o'nli xonalar
aniqligida yaxlitlandi=",B)
```



2.14-masala. Berilgan burchakni radian o‘lchovidan gradus o‘lchoviga o‘tkazing.

Yechish. A gradusga teng burchakni radian o‘lchoviga ushbu formula yordamida o‘tkaziladi. $S=\frac{A\cdot 3{,}14159}{180}$;

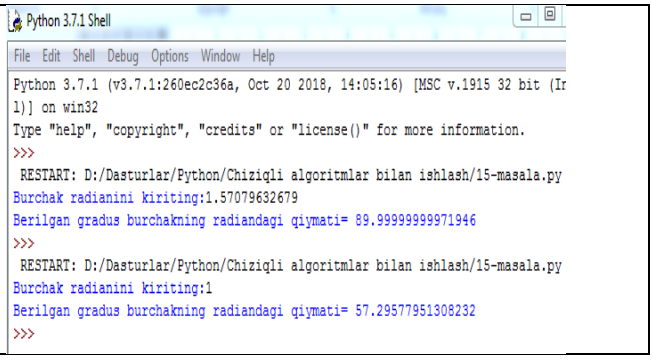
```
import math
a=float(input('Burchak gradusini kiriting:'))
s=(a*math.pi)/180
print("Berilgan radian burchakning
gradusdagi qiymati=",s)
```



2.15-masala. Berilgan burchakni gradus o‘lchovidan radian o‘lchoviga o‘tkazing.

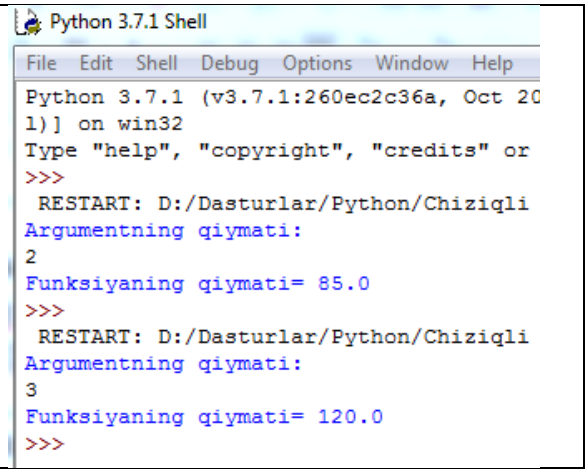
Yechish. A radianga teng burchakni gradus o‘lchoviga o‘tkazish uchun $S=\frac{A\cdot 180}{3{,}14159}$ formulasidan foydalaniladi.

```
import math
a=float(input('Burchak radianini kiriting:'))
s=(a*180)/math.pi
print("Berilgan gradus burchakning
radiandagi qiymati=",s)
```



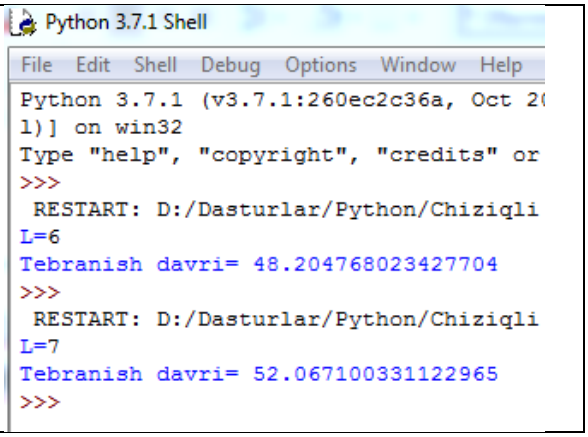
2.16-masala. Argument X ning qiymatlari berilganda $F=2(x+3)+3(x+3)^2$ funksiyaning qiymatlarini aniqlang.
Yechish. Dastur qisqaroq bo‘lishi uchun y=x+3 oraliq o‘zgaruvchi kiritamiz.

```
import math
x=float(input('Argumentning qiymati:\n'))
F=2*(x+3)+3*math.pow((x+3),2)
print("Funksiyaning qiymati=",F)
```



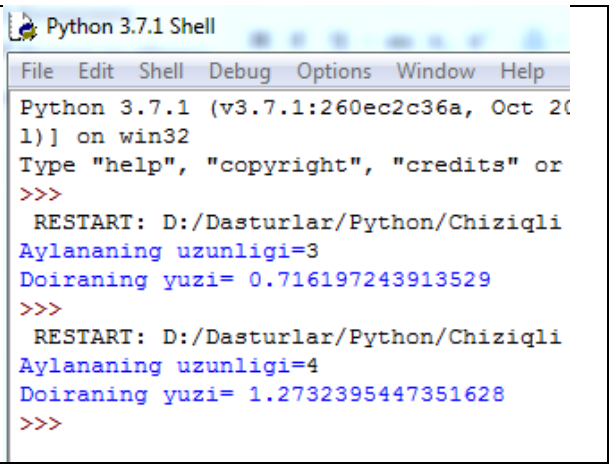
2.17-masala. Uzunligi L(m) ga teng matematik mayatnikning tebranish davrini hisoblang. (Hisoblash formulasi $T=2\pi\sqrt{LG}$, bunda $\pi=3.14; G=9.81$ (m/s^2)).

```
import math
L=float(input('L='))
G=9.81
T=2*math.pi*math.sqrt(L*G)
print("Tebranish davri=",T)
```



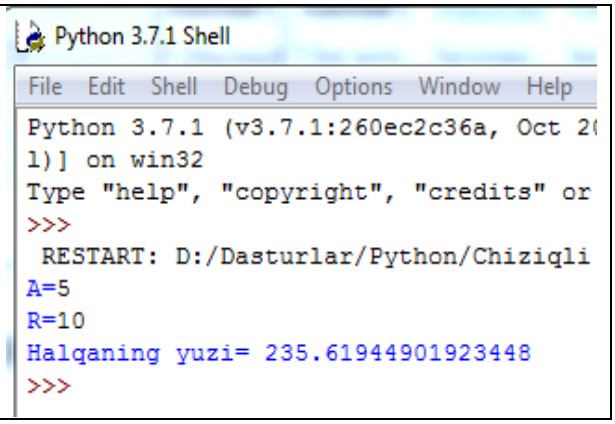
2.18-masala. Aylananing uzunligi C berilgan. Shu aylana bilan chegaralangan doiraning yuzi S ni aniqlang. (Hisoblash formulasi: $S=C^2/4\pi$).

```
import math
C=float(input("Aylananing uzunligi="))
s=math.pow(C,2)/(4*math.pi)
print("Doiraning yuzi=",s)
```



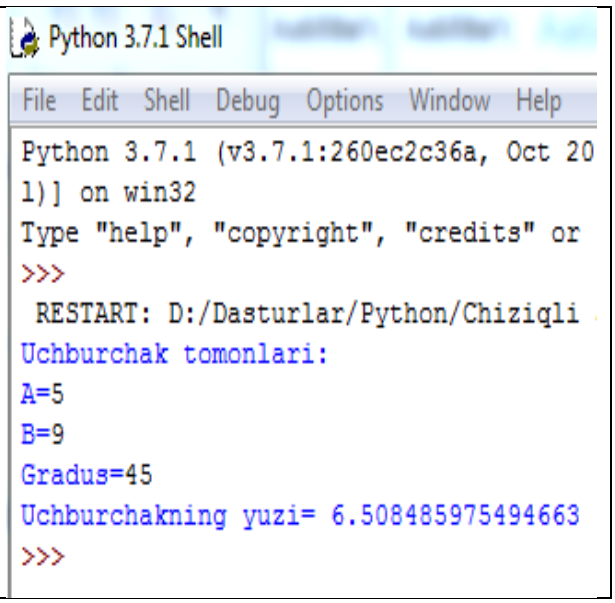
2.19-masala. Radiuslari A va R ga teng (A<R) halqa yuzi hisoblansin. (Hisoblash formulasi: $S=\pi(R^2-A^2)$).

```
import math
A=float(input('A='))
R=float(input('R='))
S=math.pi*(math.pow(R,2)-math.pow(A,2))
print("Halqaning yuzi=",S)
```



2.20-masala. Uchburchakning A va B ikkita tomoni va ular orasidagi burchagi G (gradusda) berilgan. Uchburchakning uchinchi tomonini toping. (Hisoblash formulasi:$C=\sqrt{(A^2+B^2-2AB\cdot CosG)}$).

```
import math
print("Uchburchak tomonlari:")
A=float(input("A="))
B=float(input("B="))
G=float(input("Gradus="))
G=(G*math.pi)/180
S=math.sqrt(A*A+B*B-2*A*B*math.cos(G))
print("Uchburchakning yuzi=",S)
```

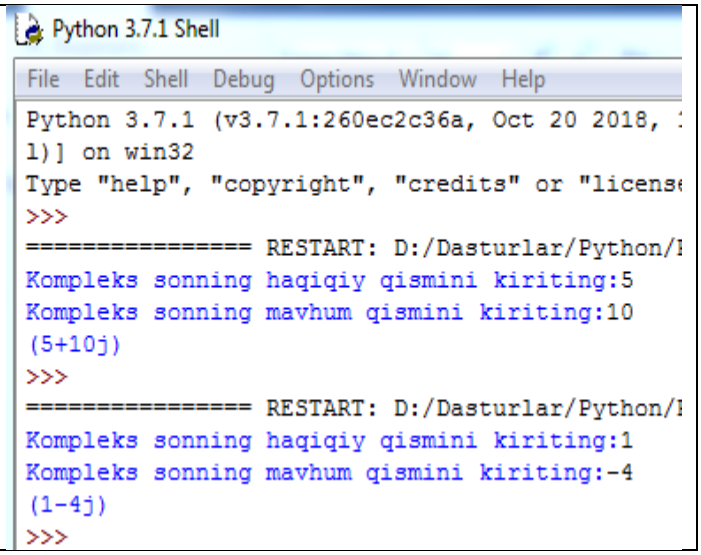


2.4. PYTHON DA KOMPLEKS SONLAR BILAN ISHLASH

Kompleks sonni yaratish uchun complex(a,b) funksiyasidan foydalanish mumkin. Bunda a - argument sifatida haqiqiy qism, b – argument sifatida, mavhum qismuzatiladi. Shuningdek, sonni a+bj ko‘rinishida ifodalanadi.

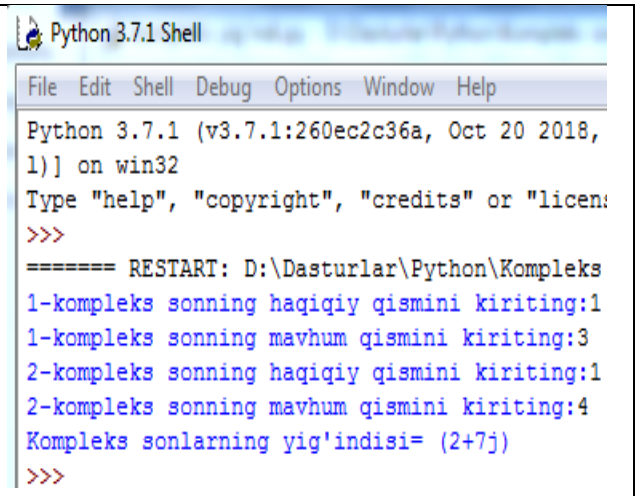
3.1-masala. Kompleks sonlarning haqiqiy va mavhum qismlarini ifodalovchi a va b sonlari berilgan. Shu sonlar orqali kompleks sonni ekranga chiqaruvchi dastur tuzing.

```
a=float(input('Kompleks sonning
haqiqiy qismini kiriting:'));
b=float(input('Kompleks sonning
mavhum qismini kiriting:'));
x=complex(a,b);
print(x);
```



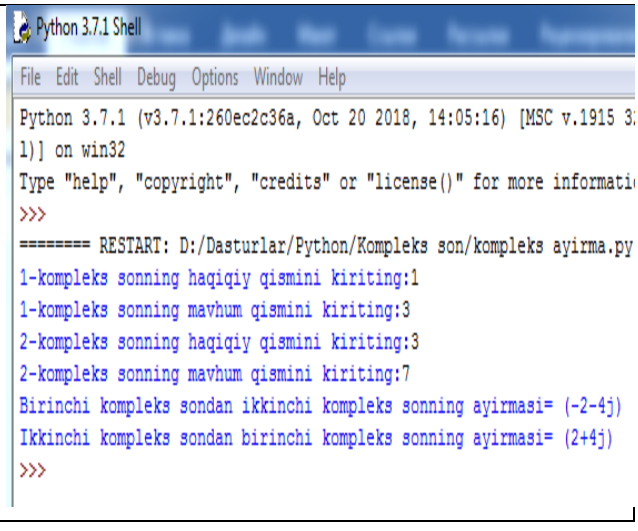
3.2-masala. Ikkita kompleks sonlarning haqiqiy va mavhum qismlari berilgan. Ushbu kompleks sonlarning yig‘indisini ekranga chiqaruvchi dastur tuzing.

```
a=float(input('1-kompleks sonning
haqiqiy qismini kiriting:'));
b=float(input('1-kompleks sonning
mavhum qismini kiriting:'));
x=float(input('2-kompleks sonning
haqiqiy qismini kiriting:'));
y=float(input('2-kompleks sonning
mavhum qismini kiriting:'));
kompleks1=complex(a,b);
kompleks2=complex(x,y);
s=kompleks1+kompleks2;
print("Kompleks sonlarning
yig'indisi=",s);
```



3.3-masala. Ikkita kompleks sonlarning haqiqiy va mavhum qismlari berilgan. Ushbu kompleks sonlarning ayirmasini ekranga chiqaruvchi dastur tuzing.

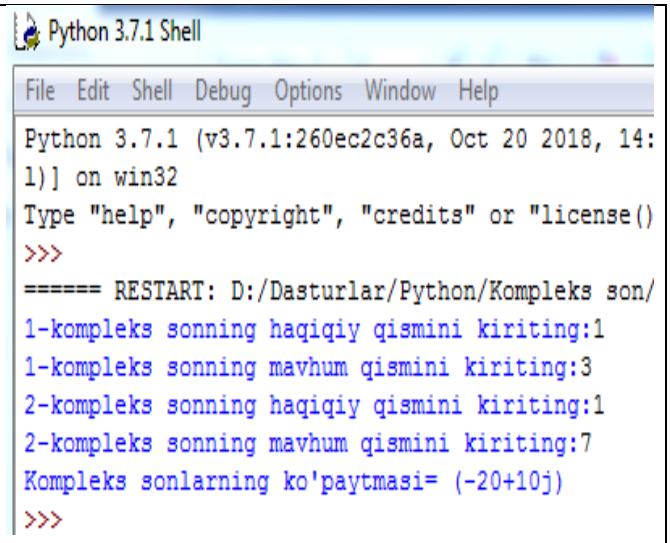
```
a=float(input('1-kompleks sonning
haqiqiy qismini kiriting:'));
b=float(input('1-kompleks sonning
mavhum qismini kiriting:'));
x=float(input('2-kompleks sonning
haqiqiy qismini kiriting:'));
y=float(input('2-kompleks sonning
mavhum qismini kiriting:'));
kompleks1=complex(a,b);
kompleks2=complex(x,y);
s1=kompleks1-kompleks2;
s2=kompleks2-kompleks1;
```



```
print("Birinchi kompleks sondan
ikkinchi kompleks sonning
ayirmasi=",s1);
print("Ikkinchi kompleks sondan
birinchi kompleks sonning
ayirmasi=",s2);
```

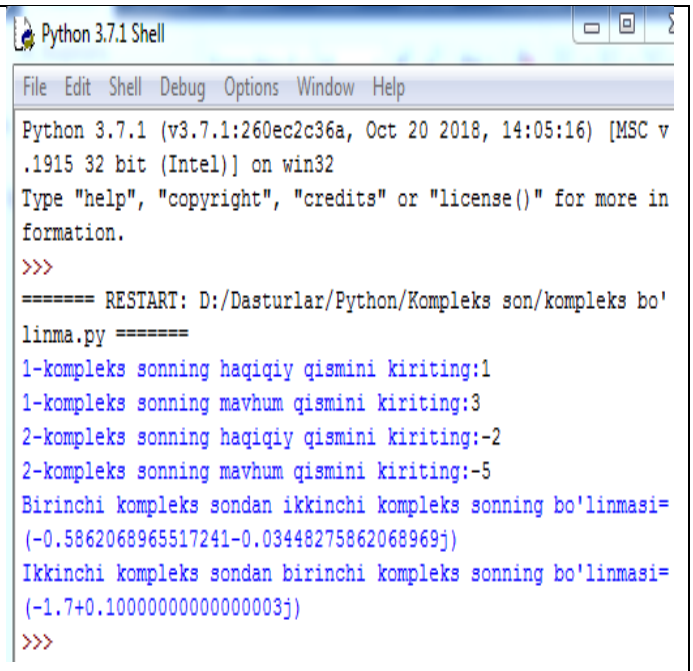
3.4-masala. Ikkita kompleks sonlarning haqiqiy va mavhum qismlari berilgan. Ushbu kompleks sonlarning ko‘paytmasini ekranga chiqaruvchi dastur tuzing.

```
a=float(input('1-kompleks sonning
haqiqiy qismini kiriting:'));
b=float(input('1-kompleks sonning
mavhum qismini kiriting:'));
x=float(input('2-kompleks sonning
haqiqiy qismini kiriting:'));
y=float(input('2-kompleks sonning
mavhum qismini kiriting:'));
kompleks1=complex(a,b);
kompleks2=complex(x,y);
s=kompleks1*kompleks2;
print("Kompleks sonlarning
ko'paytmasi=",s);
```



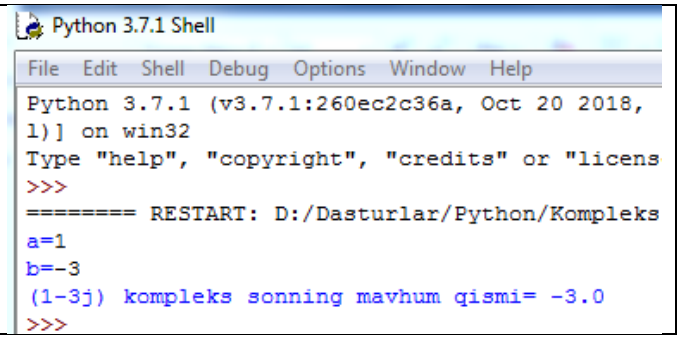
3.5-masala. Ikkita kompleks sonlarning haqiqiy va mavhum qismlari berilgan. Ushbu kompleks sonlarning bo‘linmasini ekranga chiqaruvchi dastur tuzing.

```
a=float(input('1-kompleks sonning
haqiqiy qismini kiriting:'));
b=float(input('1-kompleks sonning
mavhum qismini kiriting:'));
x=float(input('2-kompleks sonning
haqiqiy qismini kiriting:'));
y=float(input('2-kompleks sonning
mavhum qismini kiriting:'));
kompleks1=complex(a,b);
kompleks2=complex(x,y);
s1=kompleks1/kompleks2;
s2=kompleks2/kompleks1;
print("Birinchi kompleks sondan
ikkinchi kompleks sonning
bo'linmasi=",s1);
print("Ikkinchi kompleks sondan
birinchi kompleks sonning
bo'linmasi=",s2);
```



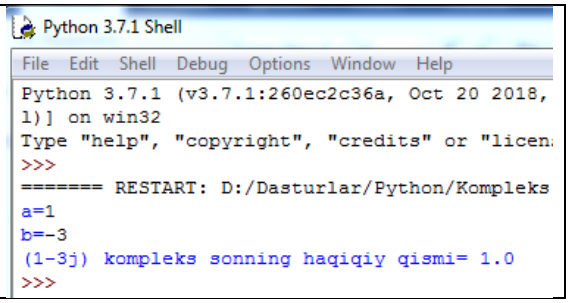
3.6-masala. Kompleks son berilgan. Ushbu kompleks sonning mavhum qismini chiqaruvchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
s1=complex(a,b);
s2=s1.imag;
print(s1,'kompleks sonning mavhum
qismi=',s2);
```



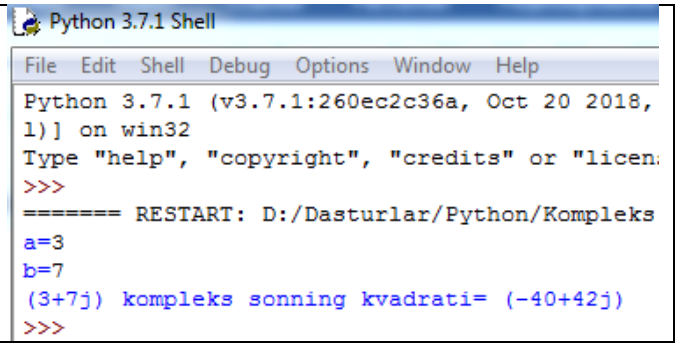
3.7-masala. Kompleks son berilgan. Ushbu kompleks sonning haqiqiy qismini chiqaruvchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
s1=complex(a,b);
s2=s1.real;
print(s1,'kompleks sonning haqiqiy
qismi=',s2);
```



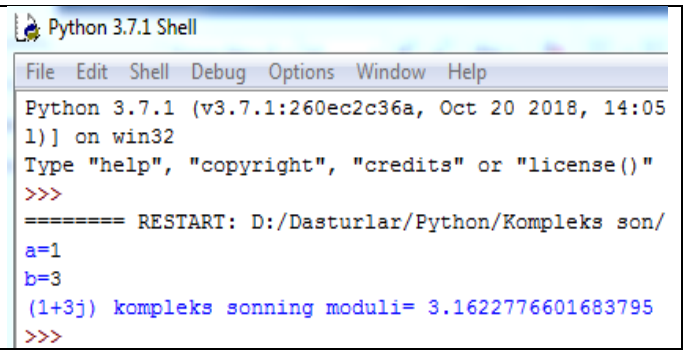
3.8-masala. Kompleks son berilgan. Ushbu kompleks sonning kvadratini hisoblovchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
s=complex(a,b);
kvadrat=pow(s,2);
print(s,'kompleks sonning
kvadrati=',kvadrat);
```



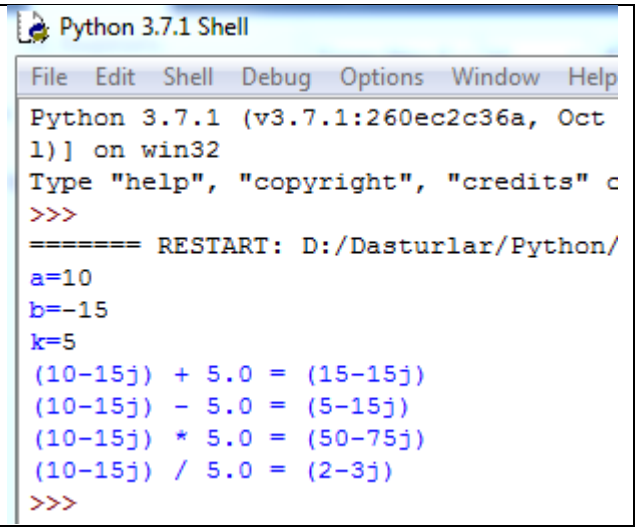
3.9-masala. Kompleks son berilgan. Ushbu kompleks sonning modulini hisoblovchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
s=complex(a,b);
modul=abs(s);
print(s,'kompleks sonning
moduli=',modul);
```



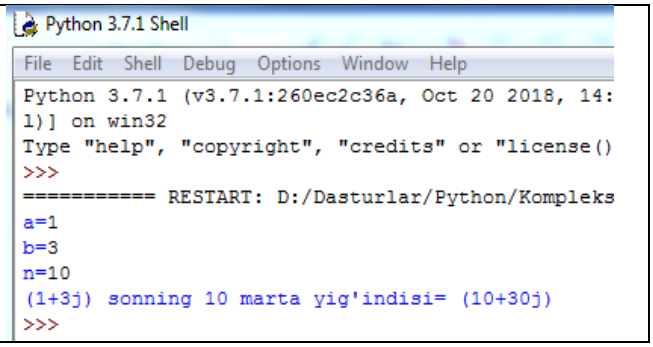
3.10-masala. Kompleks son berilgan. Ushbu kompleks songa k sonini qo‘shish, ayirish, ko‘paytirish, va bo‘lish amallarini bajaruvchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
k=float(input('k='));
s=complex(a,b);
yigindi=s+k;
ayirma=s-k;
kopaytma=s*k;
bolinma=s/k;
print(s,'+',k,'=',yigindi);
print(s,'-',k,'=',ayirma);
print(s,'*',k,'=',kopaytma);
print(s,'/',k,'=',bolinma);
```



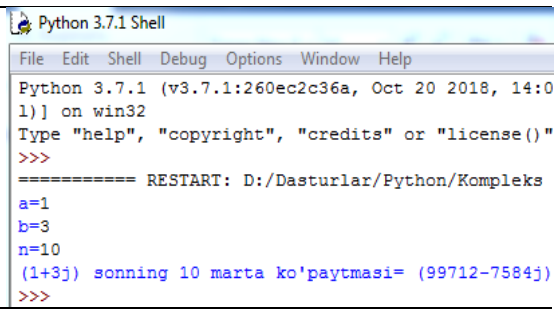
3.11-masala. Kompleks son berilgan. Ushbu sonni n marta o‘zini-o‘ziga qo‘shadigan dastur tuzilsin.

```
a=float(input('a='));
b=float(input('b='));
n=int(input('n='));
s=complex(a,b);
p=s*n;
print(s,'sonning',n,"marta yig'indisi=",p);
```



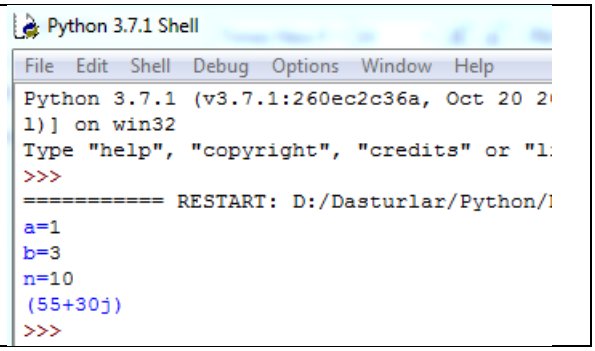
3.12-masala. Kompleks son berilgan. Ushbu sonni n marta o‘zini-o‘ziga ko‘paytiradigan dastur tuzilsin.

```
a=float(input('a='));
b=float(input('b='));
n=int(input('n='));
s=complex(a,b);
p=s**n;
print(s,'sonning',n,"marta ko'paytmasi=",p);
```



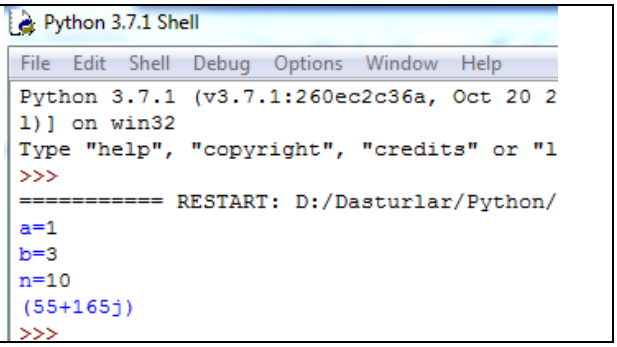
3.13-masala. Kompleks son berilgan. Ushbu songa 1 dan n gacha bo‘lgan sonlarni qo‘shib va hammasini yig‘indisini chiqaruvchi dastur tuzing.

```
a=float(input('a='));
b=float(input('b='));
n=int(input('n='));
s=complex(a,b);
p=s+(n**2+n)/2;
print(p);
```



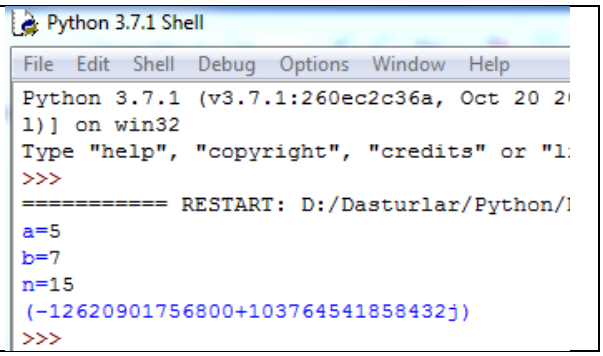
3.14-masala. Kompleks son berilgan. Ushbu songa 1 dan n gacha bo‘lgan sonlarni ko‘paytirib va hammasini yig‘indisini chiqaruvchi dastur tuzing.

```
a=float(input('a='));
b=float(input('b='));
n=int(input('n='));
s=complex(a,b);
p=s*(n**2+n)/2;
print(p);
```



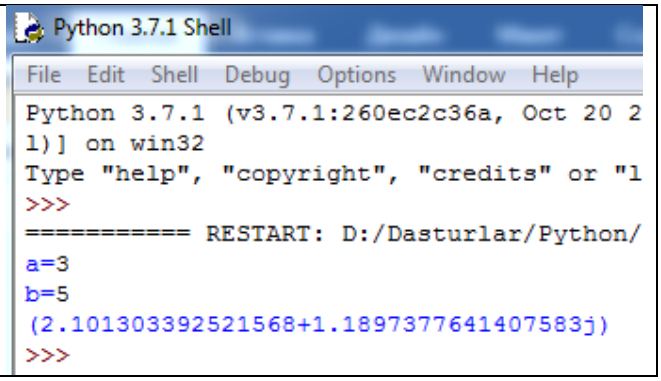
3.15-masala. Kompleks son berilgan. Ushbu sonning n darajasini chiqaruvchi dastur tuzing.

```
a=float(input('a='));
b=float(input('b='));
n=int(input('n='));
s=complex(a,b);
k=pow(s,n);
print(s,'ning',n,'-chi darajasi=',k);
```



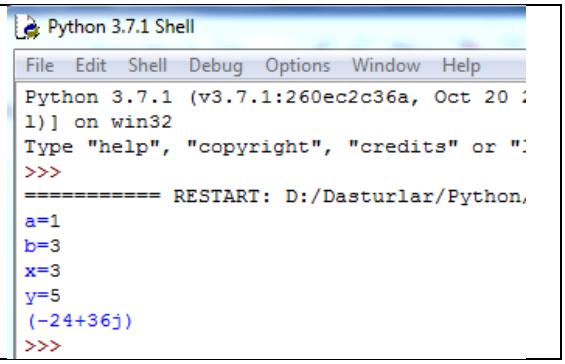
3.16-masala. Kompleks son berilgan. Ushbu kompleks sonni kvadrat ildizini topadigan dastur tuzing.

```
a=float(input('a='));
b=float(input('b='));
s=complex(a,b);
k=pow(s,1/2);
print(s,'ning kvadrat ildizi=',k);
```



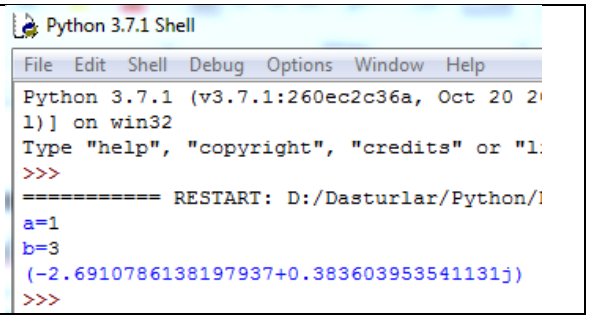
3.17-masala. Ikkita kompleks son berilgan. Ushbu kompleks sonlarning kvadratlari yig‘indisini hisoblovchi dastur tuzing.

```
a=float(input('a='));
b=float(input('b='));
x=float(input('x='));
y=float(input('y='));
s1=complex(a,b);
s2=complex(x,y);
summa=pow(s1,2)+pow(s2,2);
print(summa);
```



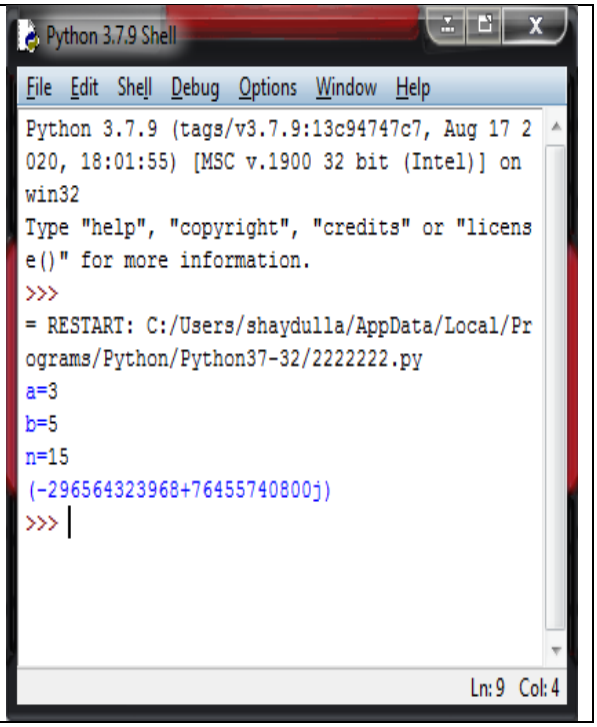
3.18-masala. Kompleks son berilgan. Ushbu kompleks son orqali e ning kompleks sondagi darasini hisoblovchi dastur tuzing.

```
import math;
a=float(input('a='));
b=float(input('b='));
s=complex(a,b);
print(pow(math.e,s));
```



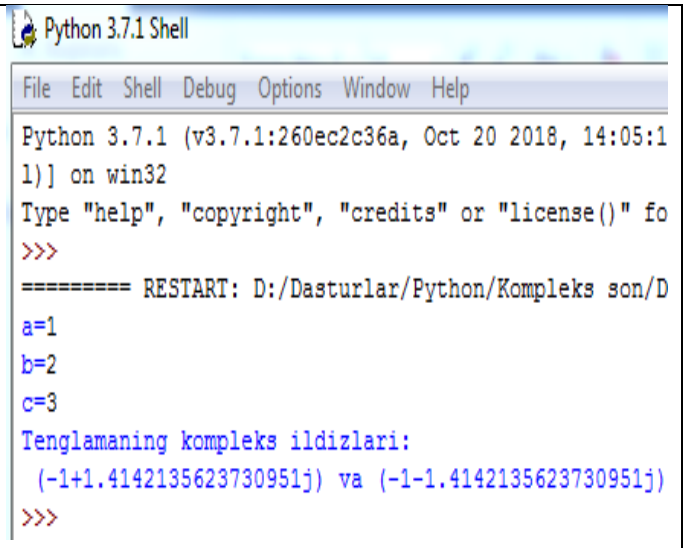
3.19-masala. Kompleks son berilgan. Ushbu kompleks son orqali 1,2,…,n ning kompleks sondagi darajasini hisoblovchi dastur tuzing.

```
a=float(input('a='));
b=float(input('b='));
n=int(input('n='));
s=complex(a,b);
p=s**n;
print(p);
```



3.20-masala. Kvadrat tenglamaning ildizlarini topuvchi dastur tuzing. Agar Diskreminant<0 bo‘lsa tenglamaning kompleks ildizlarini chiqarsin.

```
import math, cmath;
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
D=b*b-4*a*c;
if a!=0:
    if D>0:
        x1=(-b+math.sqrt(D))/(2*a);
        x2=(-b-math.sqrt(D))/(2*a);
        print("Tenglamaning
ildizlari:\n",x1,'va',x2);
    elif D<0:
        x1=(-b+cmath.sqrt(D))/(2*a);
        x2=(-b-cmath.sqrt(D))/(2*a);
        print("Tenglamaning kompleks
ildizlari:\n",x1,'va',x2);
    else:
        x=-b/(2*a);
        print("Tenglamaning ildizi:\n",x);
```



2.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Topshiriq: 1) Quyidagi matematik funksiyalarni PYTHON dasturlash tilida tuzing:

1	$a=\frac{2\cos(x-\pi/6)}{1/2+\sin^2 y}$ $b=1+\frac{z^2}{3+z^3/5}$	x = 1,426 y = -1,220, z = 3,5
2	$j=\left\lvert x^{y/x}-\sqrt[3]{y/x}\right\rvert$ $\psi=(y-x)\frac{y-z/(y-x)}{1+(y-x)^2}$	x = 1,825 y = 18,225 z = -3,298
3	$S=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}$, $\varphi=x(\sin x^3+\cos^2 y)$	x = 0,335 y = 0,025
4	$y=e^{-bt}\sin(at+b)-\sqrt{\lvert bt+a\rvert}$, $S=b\sin(at^2\cos 2t)-1$	a = -0,5, b = 1,7 t = 0,44
5	$\omega=\sqrt{x^2+b}-b^2\sin^3(x+a)/x$ $y=\cos^2 x^3-x/\sqrt{a^2+b^2}$	a = 1,5 b = 15,5 x = -2,9
6	$S=x^3 tg^2(x+b)^2+a/\sqrt{x+b}$, $Q=\frac{bx^2-a}{Ax}$	a = 16,5, b = 3,4 j = 0,61

7	$R = x^2(x+1)/b - \sin^2(x+a)$, $S = \sqrt{xb/a} + \cos^2(x+b)^3$	a = 0,7, b = 0,05 x = 0,5
8	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x+b)^2$	a = 1,1, b = 0,004 x = 0,2
9	$f = \sqrt[3]{mtgt + \lvert c\sin t\rvert}$, $z = m\cos(bt\sin t) + c$	m = 2, c = -1 t = 1,2, b = 0,7
10	$y = btg^2 x - \frac{A}{\sin^2(x/a)}$, $S = b\sin(at^2\cos 2t) - 1$	a = 3,2 b = 17,5, x = -4,8
11	$a = \frac{2\cos(x - \pi/6)}{1/2 + \sin^2 y}$, $b = 1 + \frac{z^2}{3 + z^3/5}$	x = 1,4 y = -1,2, z = 3,05
12	$j = \left\lvert x^{y/x} - \sqrt[3]{y/x}\right\rvert$, $\psi = (y-x)\frac{y - z/(y-x)}{1+(y-x)^2}$	x = 1,8 y = 18,2 z = -3,02
13	$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$, $\varphi = x(\sin x^3 + \cos^2 y)$	x = 0,303 y = 0,02
14	$y = e^{-bt}\sin(at+b) - \sqrt{\lvert bt+a\rvert}$, $S = b\sin(at^2\cos 2t) - 1$	a = -0,05, b = 1,17 t = 0,24
15	$\omega = \sqrt{x^2+b} - b^2\sin^3(x+a)/x$, $y = \cos^2 x^3 - x/\sqrt{a^2+b^2}$	a = 1,15 b = 15,05 x = -2,19
16	$S = x^3 tg^2(x+b)^2 + a/\sqrt{x+b}$, $Q = \frac{bx^2 - a}{Ax}$	a = 1,5, b = 3,14 j = 0,65
17	$R = x^2(x+1)/b - \sin^2(x+a)$, $S = \sqrt{xb/a} + \cos^2(x+b)^3$	a = 0,17, b = 0,5 x = 0,15
18	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}$, $Z = \frac{x^2}{A} + \cos(x+b)^2$	a = 1,01, b = 0,04 x = 0,12
19	$f = \sqrt[3]{mtgt + \lvert c\sin t\rvert}$, $z = m\cos(bt\sin t) + c$	m = 2, c = -1 t = 1,02, b = 0,17
20	$y = btg^2 x - \frac{A}{\sin^2(x/a)}$, $S = b\sin(at^2\cos 2t) - 1$	a = 3,02, b = 17,15 x = -4,28

Topshiriq: 1) Quyidagi mantiqiy masalalarni PYTHON dasturlash tilida tuzing:

1.1-masala. A butun soni berilgan. Jumlani rostlikka tekshiring: “A soni musbat”.

1.2-masala. A butun soni berilgan. Jumlani rostlikka tekshiring: “A soni juft son”.

1.3-masala. Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshiring: “A>=0 yoki B<-2”

1.4-masala. Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshiring: “A va B sonlarning hech bo‘lmaganda bittasi toq son”.

1.5-masala. Ikkita butun A va B sonlari berilgan. Jumlani rostlikka tekshiring: “A va B sonlarining har ikkalasi ham yoki toq son yoki juft son”.

1.6-masala. Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshiring: “A, B, C sonlarning hech bo‘lmaganda bittasi musbat”.

1.7-masala. Uchta A, B, C butun sonlar berilgan. Jumlani rostlikka tekshiring: “A, B, C sonlardan faqat ikkitasi musbat son”.

1.8-masala. Jumlani rostlikka tekshiring: “Berilgan uchta butun sonlarning hech bo‘lmaganda bir jufti o‘zaro qarama-qarshi”.

1.9-masala. Uch xonali son berilgan. Jumlani rostlikka tekshiring: “Ushbu sonning raqamlari ketma - ket o‘suvchi bo‘lib joylashgan”.

1.10-masala. Uch xonali son berilgan. Jumlani rostlikka tekshiring: “Ushbu sonning raqamlari ketma - ket o‘suvchi bo’lib joylashgan yoki kamayuvchi ketma - ketlikka ega”.

1.11-masala. Uch xonali son berilgan. Jumlani rostlikka tekshiring: “Ushbu sonni chapdan o‘qiganda ham, o‘ngdan o‘qiganda ham bir xil”.

1.12-masala. x, y sonlar berilgan. Jumlani rostlikka tekshiring: “Koordinatalari (x,y) bo‘lgan nuqta koordinata choragining to‘rtinchisida yotadi”.

1.13-masala. x, y sonlar berilgan. Jumlani rostlikka tekshiring: “Koordinatalari (x,y) bo‘lgan nuqta koordinata choragining ikkinchisida yoki uchunchisida yotadi”.

1.14-masala. x, y sonlar berilgan. Jumlani rostlikka tekshiring: “Koordinatalari (x,y) bo’lgan nuqta koordinata choragining birinchi yoki uchunchisida yotadi”.

1.15-masala. a, b, c butun sonlari berilgan. Jumlani rostlikka tekshiring: “a, b, c tomonli uchburchak teng yonli bo‘ladi”.

1.16-masala. a, b, c butun sonlar berilgan. Jumlani rostlikka tekshiring: “a, b, c tomonli uchburchak to‘g‘ri burchakli”.

1.17-masala. Shaxmat doskasining ikkita turli (x1, y1), (x2, y2) koordinatalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Jumlani rostlikka tekshiring: “Berilgan maydonlar bir xil rangda”.

1.18-masala. Shaxmat doskasining ikkita turli (x1, y1), (x2, y2) koordinatalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Jumlani rostlikka tekshiring: “Shoh bir yurishda bir maydondan ikkinchisiga o‘ta oladi”.

1.19-masala. Shaxmat doskasining ikkita turli (x1, y1), (x2, y2) koordinatalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Jumlani rostlikka tekshiring: “Fil bir yurishda bir maydondan ikkinchisiga o‘ta oladi”.

1.20-masala. Shaxmat doskasining ikkita turli (x1, y1), (x2, y2) koordinatalari berilgan (1-8 oraliqda yotuvchi butun sonlar). Jumlani rostlikka tekshiring: “Farzin bir yurishda bir maydondan ikkinchisiga o‘ta oladi”.

Topshiriq: 2) Quyidagi chiziqli masalalarni PYTHON dasturlash tilida tuzing:

2.1-masala. Kvadratning tomoni a berilgan. Uning perimetri P=4*a va yuzasi $S=a^2$ ni hisoblash dasturini tuzing.

2.2-masala. Kubning yon tomoni a berilgan. Uning hajmini $V = a^3$ va to‘la sirti $S=6*a^2$ ni hisoblash dasturini tuzing.

2.3-masala. Paralelepepidning tomonlari a, b, c berilgan. Uning hajmini V = a*b*c va to‘la sirti S = 2*(a*b+b*c+a*c) ni hisoblash dasturini tuzing.

2.4-masala. Nolga teng bo'Imagan ikkita son berilgan. Ularning yig‘indisini, ko‘paytmasini va har birining modulini hisoblash dasturini tuzing.

2.5-masala. Umumiy markazga ega bo‘lgan ikkita aylana radiusi berilgan: R_1,R_2 ($R_1>R_2$).Ularnig yuzalari S_1 va S_2, ularning ayirmasi S_3 ni hisoblash dasturini tuzing. $S_1 = \pi R_1^2$, $S_2= \pi R_2^2$, $S_3 = \pi (R_1^2 - R_2^2)$.

2.6-masala. Aylananing uzunligi L berilgan. Uning radiusi R va yuzasi S ni hisoblang dasturini tuzing. $L = 2* \pi *R$, $S = \pi R^2$, $\pi = 3.14$.

2.7-masala. Aylananing yuzasi S berilgan. Uning diametri D va radiusi R ni hisoblash dasturini tuzing. $L = 2* \pi *R$ $S = \pi R^2$ $\pi = 3.14$

2.8-masala. Sonlar o‘qida A, B, C nuqtalar berilgan. AC va BC kesmalarning uzunligini va kesmalar uzunligining yig‘indisini hisoblash dasturini tuzing.

2.9-masala. Sonlar o‘qida A, B, C nuqtalar berilgan. C nuqta A va B nuqtalar orasida joylashgan. AC va BC kesmalar uzunligining ko‘paytmasini toping va dasturini tuzing.

2.10-masala. To‘g‘ri to‘rtburchakning qarama-qarshi uchlari koordinatlari berilgan. Uning tomonlari koordinata o‘qiga parallel. To‘g‘ri to‘rtburchakning perimetri va yuzasini hisoblash dasturini tuzing.

2.11-masala. Uchburchakning uchta tomoni uchlari koordinatalari berilgan: (x1,y1), (x2,y2), (x3,y3). Ikki nuqta orasidagi masofani toping dasturini tuzing.

2.12-masala. A, B va C sonlari berilgan. A ning qiymati B ga, B ning qiymati C ga va C ning qiymati A ga almashtirilsin. A, B va C ning yangi qiymatlarini ekranga chiqaruvchi dastur tuzing.

2.13-masala. x ning qiymati berilganda $y=3x^6-6x^2-7$ funksiyaning qiymatini hisoblash dasturini tuzing.

2.14-masala. x ning qiymati berilganda $y=4(x-3)^6-7(x-3)^3+2$ funksiyaning qiymatini hisoblash dasturini tuzing.

2.15-masala. A soni berilgan. A ning A^2, A^3, A^5 ,A^{10}, A^{15} darajalarini aniqlovchi dastur tuzing.

2.16-masala. Temperatura T_F Farengeytda berilgan. Temperatura qiymatini T_C gradus selsiyga o‘tkazuvchi dastur tuzing: $T_C=(T_F-32)*5/9$.

2.17-masala. Temperatura T_C gradus selsiyda berilgan. Temperatura qiymatini T_F Farengeytga o‘tkazuvchi dastur tuzing: $T_C=(T_F-32)*5/9$.

2.18-masala. X kg shokolad A so‘m turadi va Y kg konfet B so‘m turadi. 1 kg shokolad 1 kg konfetdan qancha qimmat turishini aniqlovchi dastur tuzing.

2.19-masala. Qayiqning tezligi V km/soat, daryo oqimining tezligi U km/soat (V>U) Qayiqning daryo oqimi bo‘yicha xarakatlanish vaqti T1, oqimga qarshi T2 Qayiqni yurgan S yo‘lini aniqlovchi dastur tuzing.

2.20-masala. Birinchi avtomabilning tezligi V1 km/soat, ikkinchisiniki V2 km/soat, ular orasidagi masofa S km. Ular biri-biri tomonga harakatlana boshlasa T vaqtdan keyin ular orasidagi masofani aniqlaydigan dastur tuzing.

Topshiriq: 2) Quyidagi kompleks masalalarni PYTHON dasturlash tilida tuzing:

3.1-masala. $z_1 = 1 + \sqrt{3}j$, $z_2 = 1 - \sqrt{3}j$ kompleks sonlar berilgan. Ushbu $z_1 \cdot z_2 = ?,\ z_1 + z_2 = ?,\ z_1 - z_2 = ?,\ \frac{z_1}{z_2} = ?$ amallarni bajaruvchi dastur tuzing.

3.2-masala. $z = \frac{1}{(1-\sqrt{3}j)^6}$ ifodani bajaruvchi dastur tuzing.

3.3-masala. $z = (1+\sqrt{3}j)^{15}$ ifodani bajaruvchi dastur tuzing.

3.4-masala. $(-1)^{\sqrt{3}}$ ifodani bajaruvchi dastur tuzing.

3.5-masala. $z_1 = 3j,\ z_2 = -\sqrt{3} + j$ kompleks sonlar berilgan. $\frac{-z_1}{z_2}$ ni hisoblovchi dastur tuzing.

3.6-masala. $z_1 = 3j,\ z_2 = -\sqrt{3} + j$ kompleks sonlar berilgan. $\left(\frac{-z_2 - z_1 j}{2z_2}\right)^2$ ni hisoblovchi dastur tuzing.

3.7-masala. $z_1 = 3j,\ z_2 = -\sqrt{3} + j$ kompleks sonlar berilgan. z_1^6, z_2^6 ni hisoblovchi dastur tuzing.

3.8-masala. $z_1 = 3j,\ z_2 = -\sqrt{3} + j$ kompleks sonlar berilgan. $\sqrt[3]{z_1}, \sqrt[3]{z_2}$ ni hisoblovchi dastur tuzing.

3.9-masala. $\left(\cos\frac{\pi}{6} + j \cdot \sin\frac{\pi}{6}\right)^6$ ni hisoblovchi dastur tuzing.

3.10-masala. $\left(\frac{3}{2} - \frac{\sqrt{3}}{2}j\right)^{10}$ ni hisoblovchi dastur tuzing.

3.11-masala. $\left(\cos 35^\circ + j \cdot \sin 35^\circ\right)^{-12}$ ni hisoblovchi dastur tuzing.

3.12-masala. $\sqrt[4]{-2 + 2\sqrt{3}j}$ ni hisoblovchi dastur tuzing.

3.13-masala. Quyidagi kompleks sonlarning ko‘paytmasi va bo‘linmasini toping:

$$z_1 = 10\left(\cos\frac{3\pi}{4} + j\cdot\sin\frac{3\pi}{4}\right),\ z_2 = 2\left(\cos\frac{\pi}{4} + j\cdot\sin\frac{\pi}{4}\right)$$

3.14-masala. Quyidagi kompleks sonlarning ko‘paytmasi va bo‘linmasini toping:

$$z_1 = 6\left(\cos\frac{\pi}{2} + j\cdot\sin\frac{\pi}{2}\right),\ z_2 = \cos\frac{\pi}{6} + j\cdot\sin\frac{\pi}{6}$$

3.15-masala. Quyidagi kompleks sonlarning ko‘paytmasi va bo‘linmasini toping:

$$z_1 = 4\left(\cos 150^\circ + j\cdot\sin 150^\circ\right),\ z_2 = \cos(-120^\circ) + j\cdot\sin\left(-120^\circ\right)$$

3.16-masala. $z_1 = 3 + j,\ z_2 = 2j$ kompleks sonlar berilgan. $\frac{z_2}{-z_1}$ ni hisoblovchi dastur tuzing.

3.17-masala. $z_1 = 3 + j,\ z_2 = 2j$ kompleks sonlar berilgan. $\left(\frac{z_1 + z_2}{-3z_2}\right)^8$ ni hisoblovchi dastur tuzing.

3.18-masala. $z_1 = 3 + j,\ z_2 = 2j$ kompleks sonlar berilgan. $\left(\frac{z_1^2 + z_2}{2z_2}\right)^4$ ni hisoblovchi dastur tuzing.

3.19-masala. $z_1 = 3 + j,\ z_2 = 2j$ kompleks sonlar berilgan. $\left(\frac{-z_1^2 + z_2^2}{z_1 z_2}\right)^6$ ni hisoblovchi dastur tuzing.

3.20-masala. $\sqrt[3]{-1}$ ni hisoblovchi dastur tuzing.

III. BOB. PYTHON DA TARMOQLANUVCHI OPERATORLAR

3.1. PYTHON DA SHART OPERATORI IF...ELSE

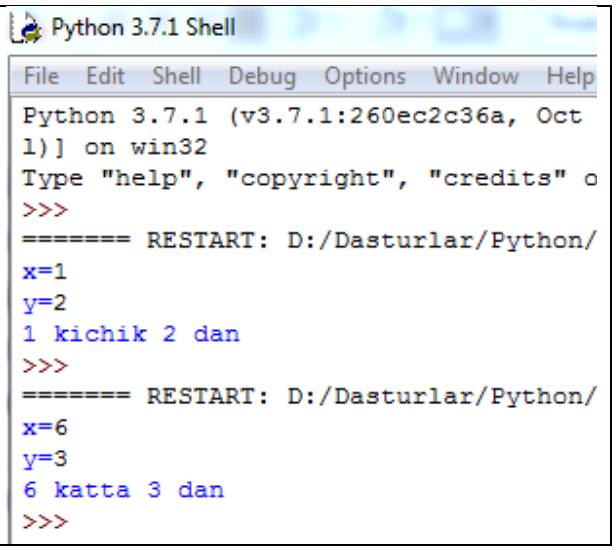
Bu operator **PYTHON** dasturlash tilidagi muhim operatorlardan biridir. U shartga bog‘liq ravishda kod fragmentini bajarishga mo‘ljallangan. Shart operatori boshqarishni qaysi tarmoqqa uzatishni ta’minlaydi. Shart operatorining umumiy ko`rinishi:

if <shart>
    <operator1>
else
    <operator2>

Shartli operator sintaksisi: ***if*** (<shart>) <operator1> ***else*** <operator2>. Shart **<shart>** ixtiyoriy shartli ifoda bo‘lishi mumkin. Agar u rost bo‘lsa **operator1** bajariladi. Aks xolda **operator2** bajariladi. Bu ixtiyoriy murakkablikdagi tekshirishlar ketma ketligini hosil qilishga imkon beradi. Bu ketma - ketlikda shartli operator to‘la yoki qisqa shaklda bo‘lishi mumkin. Shuning uchun ***if*** va ***else*** operatorlarini bir - biriga mos qo‘yishda xatolik kelib chiqishi mumkin. Tilning sintaksisi bo‘yicha ichki joylashtirilgan shartli operatorlarda har bir ***else*** *eng yaqin* ***if*** *ga mos keladi.*

Agar **x** teng **1** va **y** teng **2** bo‘lsa ***x kichik y dan*** *jumla ekranga chiqariladi,* chunki ***else*** eng yaqin ***if*** *ga mos keladi*.

```
x=int(input('x='))
y=int(input('y='))
if x<y:
   print(x,'kichik',y,'dan')
else: print(x,'katta',y,'dan')
```



3.2. PYTHON DA BIR NECHTA SHARTLARNI TEKSHIRISH IF-ELIF-ELSE OPERATORI

If yordamida biz faqatgina bitta shartni tekshira olamiz va uning natijasiga ko’ra (True/False) dasturimiz ma’lum bir amallarni bajaradi. Agar dastur davomida bir nechta shartlarni tekshirish talab qilinsa , if-elif-else ketma-ketligidan foydalanamiz. Bu ketma-ketlikning umumiy ko’rinishi quyidagicha:

if <shart1>
<operator1>;
elif <shart2>
<operator2>**;**
…

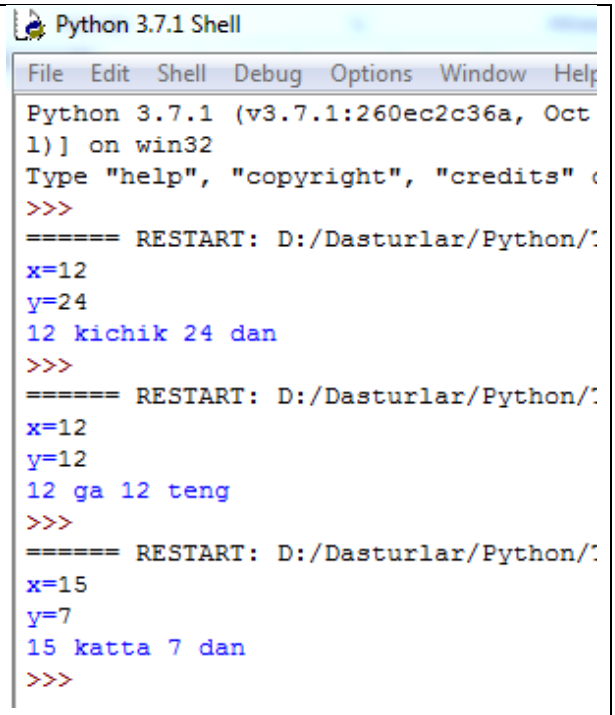
elif <shartN>
<operatorN>**;**
else

<operatorN+1>

if-elif-else ketma-ketligida Python avval if <shart1> ni tekshiradi, shart bajarilmasa, keyingi elif ga o'tadi, birinchi elif sharti bajarilmasa, keyingi elif ga o'tadi va hokazo davom etaveradi.

Misol uchun x va y sonlari kiritilganda ularni bir-biri bilan taqqoslaydigan dastur va uning natijasini ko'rib chiqaylik:

```
x=int(input('x='))
y=int(input('y='))
if x<y:
   print(x,'kichik',y,'dan')
elif x==y:
   print(x,'ga',y,'teng')
else: print(x,'katta',y,'dan')
```



1-holatda: x ga 12, y ga 24 qiymatlarini berganimizda natijamiz: "12 kichik 24 dan" javobi chiqadi,

2-holatda: x ga 12, y ga 12 qiymatlarini berganimizda natijamiz: "12 ga 12 teng" javobi chiqadi,

3-holatda: x ga 15, y ga 7 qiymatlarini berganimizda natijamiz: "15 katta 7 dan" javoblari chiqadi.

3.3. PHP DA IF…ELSE VA ELIF OPERATORI TADBIQI

4.1-masala. $Ax^2+Bx+C=0$ kvadrat tenglamaning ildizlarini toping.
Yechish. Kiritiladigan ma'lumotlar – bu tenglama koeffitsienti: a – noma'lumning ikkinchi darajasi oldidagi koeffisient; b – noma'lumning birinchi darajasi oldidagi koeffisient; c – ozod had.
Topiladigan natija – x1 va x2 tenglama ildizlari.
Buyruqlar: Diskriminantni hisoblash formulasi: $d=b^2-4ac$
Agar diskirminant natijasi noldan katta bo'lsa, u xolda quyidagi formula bilan tenglama ildizlari topiladi:

$$x1=\frac{-b-\sqrt{d}}{2a};$$

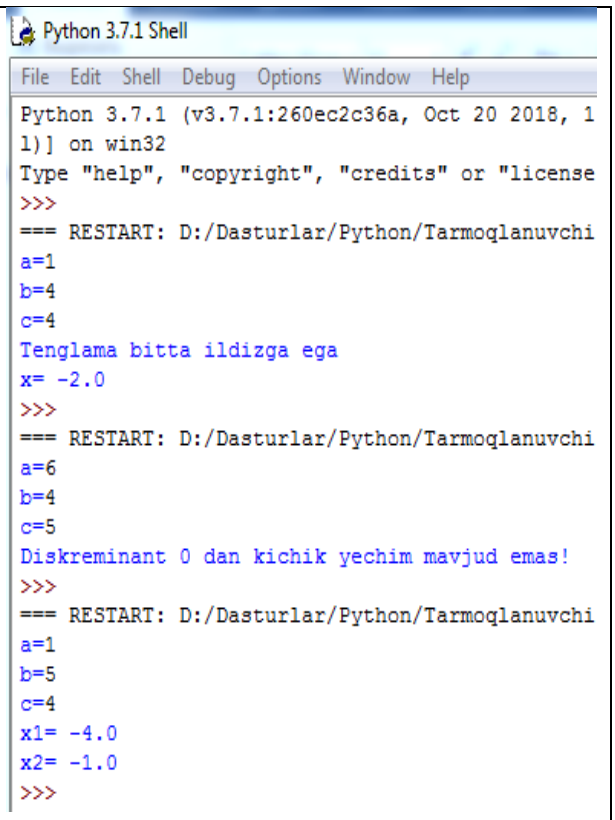
$$x2=\frac{-b+\sqrt{d}}{2a}$$

Agar diskerminant natijasi nolga teng bo'lsa, u xolda quyidagi formula bilan tenglama ildizlari topiladi:

$$x1=\frac{-b}{2a}$$

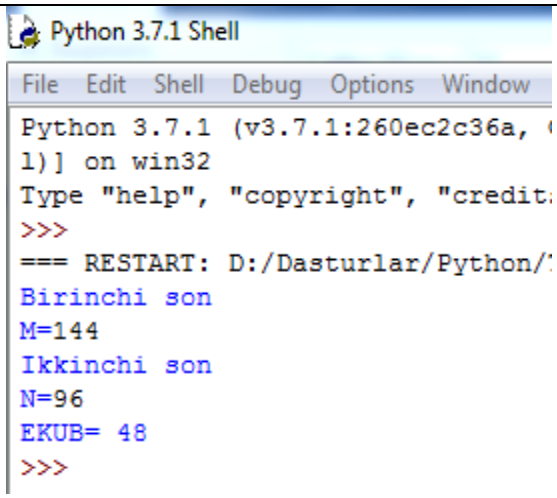
Agar diskerminant natijasi noldan kichik bo'lsa, bu tenglamaning haqiqiy ildizi yo'qligini bildiradi.

```
import math
a=float(input('a='))
b=float(input('b='))
c=float(input('c='))
d=math.pow(b,2)-4*a*c
if d>0:
    x1=(-b-math.sqrt(d))/(2*a)
    x2=(-b+math.sqrt(d))/(2*a)
    print('x1=',x1,'\nx2=',x2)
elif d==0:
    x=-b/(2*a)
    print('Tenglama bitta ildizga ega\nx=',x)
else: print('Diskreminant 0 dan kichik yechim
mavjud emas!')
```



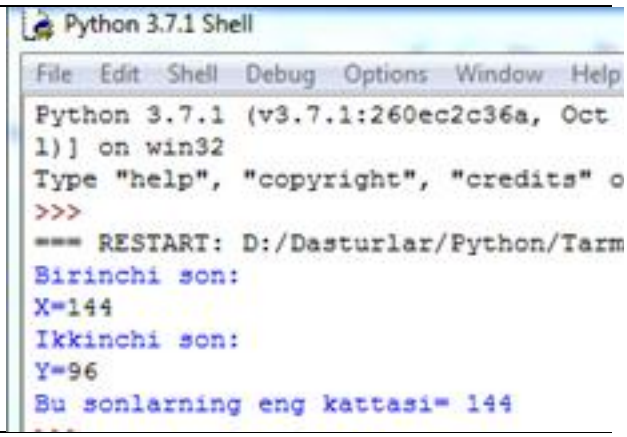
4.2-masala. Ikki butun musbat son M va N larning eng katta umumiy bo‘luvchisi (EKUB) ni aniqlang.

```
M=int(input('Birinchi son\nM='))
N=int(input('Ikkinchi son\nN='))
while M!=N:
    if M>N: M-=N
    else: N-=M
print("EKUB=",M)
```



4.3-masala. Ikkita X va Y sonlarning kattasini tanlash (EKT) dasturini tuzing.

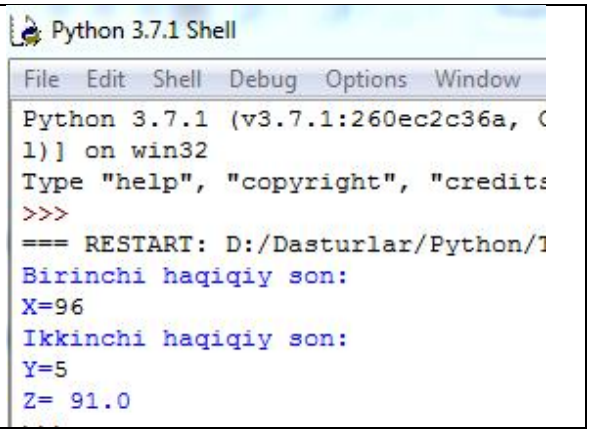
```
X=int(input('Birinchi son:\nX='))
Y=int(input('Ikkinchi son:\nY='))
if X>Y:
    print("Bu sonlarning eng kattasi=",X)
elif X==Y:
    print("Bu sonlar bir-biriga teng!")
else:
    print("Bu sonlarning eng kattasi=",Y)
```



4.4-masala. X va Y haqiqiy sonlar berilgan. Z ni hisoblang:

$$Z=\begin{cases} X-Y,\ agar\ \ X>Y\ \ bo'lsa \\ X+1,\ agar\ \ X\le Y\ \ bo'lsa \end{cases}$$

```
X=float(input('Birinchi haqiqiy son:\nX='))
Y=float(input('Ikkinchi haqiqiy son:\nY='))
if X>Y:
    Z=X-Y
    print("Z=",Z)
elif X<=Y:
    Z=X+1
    print("Z=",Z)
```



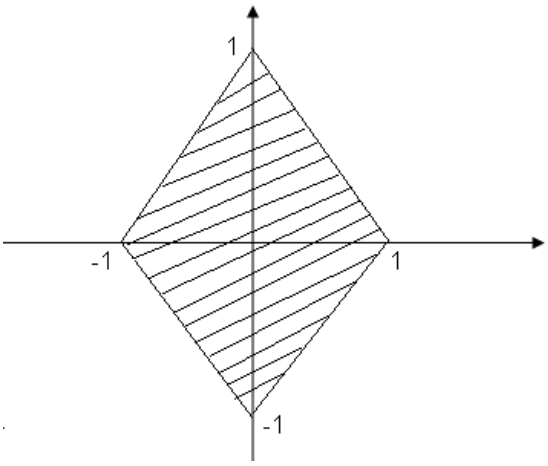
4.5-masala. Raketa ϑ (km/soat) tezlik bilan yer ekvatoridagi nuqtadan yerning quyosh atrofidagi orbitasi bo‘ylab uchiriladi. Raketani uchirish natijasi qanday bo‘ladi?. Yechish. Ma’lumki, agar $\vartheta < 7{,}8\ \frac{\text{km}}{\text{s}}$; bo‘lsa, raketa yerga qaytib tushadi. Agar $7{,}8 < \vartheta < 11{,}2$ bo‘lsa, raketa yer yo‘ldoshiga aylanadi; Agar $11{,}2 < \vartheta < 16{,}4$ bo‘lsa, raketa quyosh yo‘ldoshiga aylanadi; Agar $\vartheta > 16{,}4$ bo‘lsa, raketa quyosh sistemasidan chiqib ketadi.

```
V=float(input("Raketa tezligi\nV="))
if V<7.8:
    print("Raketa yerga qaytib tushadi")
if V>7.8 and V<11.2:
    print("Raketa yer yo'ldoshiga aylanadi")
if V>11.2 and V<16.4:
    print("Raketa quyosh yo'ldoshiga aylanadi")
if V>16.4:
    print("Raketa quyosh sistemasidan chiqib
ketadi")
```

```
Python 3.7.1 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14
1)] on win32
Type "help", "copyright", "credits" or "license
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
Raketa tezligi
V=20
Raketa quyosh sistemasidan chiqib ketadi
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
Raketa tezligi
V=15
Raketa quyosh yo'ldoshiga aylanadi
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
Raketa tezligi
V=10
Raketa yer yo'ldoshiga aylanadi
>>>
=== RESTART: D:/Dasturlar/Python/Tarmoqlanuvchi
Raketa tezligi
V=5
Raketa yerga qaytib tushadi
>>>
```

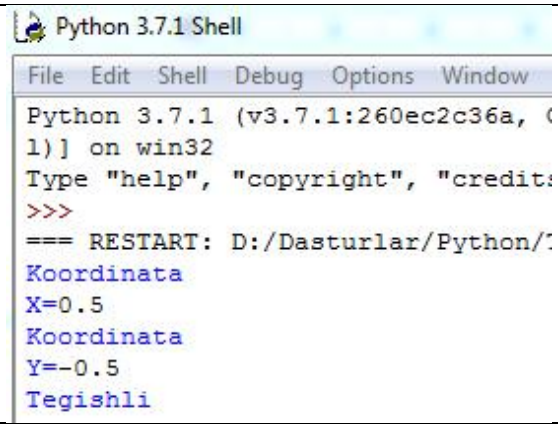
4.6-masala. Koordinatalari x va y ga teng bo‘lgan nuqta 14-rasmda tasvirlangan tekislikdagi shaklga tegishlimi?

Yechish. Koordinatalari quyidagi shatrlarni qanoatlantiradigan nuqtalar berilgan shaklga tegishli bo‘ladi: $|x| + |y| \le 1$



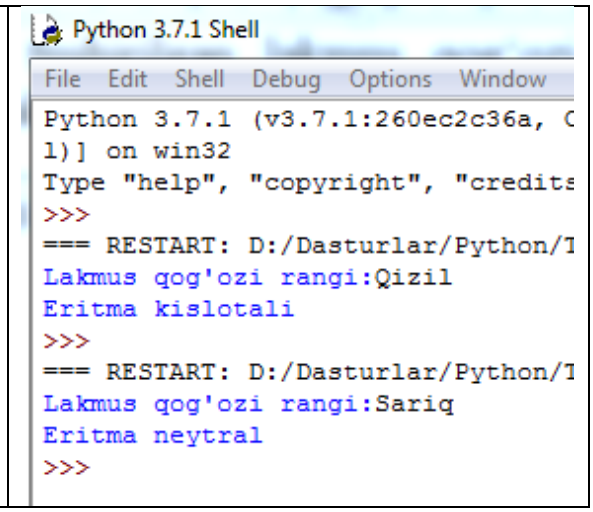
14-Rasm

```
X=float(input('Koordinata\nX='))
Y=float(input('Koordinata\nY='))
if abs(X)+abs(Y)<=1:
    print('Tegishli')
else:
    print('Tegishli emas')
```



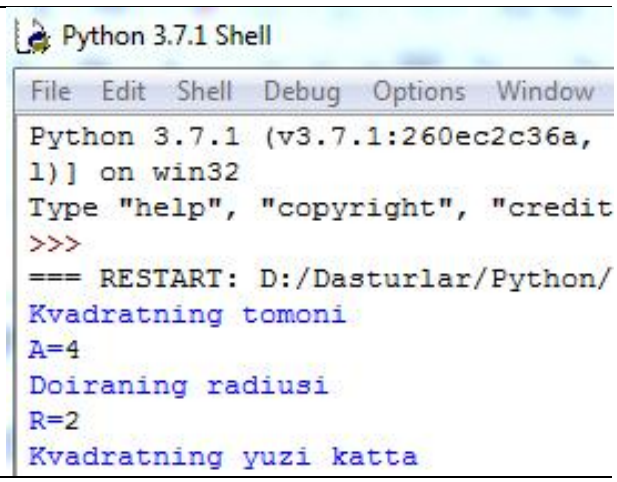
4.7-masala. Lakmus qog‘ozidan foydalanib eritma muhitini aniqlang.
Yechish. Ma’lumki, eritmaga tushirilgan lakmus qog‘ozi qizil bo‘lsa, eritma kislotali; Ko‘k bo‘lsa, ishqorli; aks holda eritma neytral bo‘ladi.

```
A=input("Lakmus qog'ozi rangi:")
if A=="Qizil":
    print("Eritma kislotali")
elif A=="Ko'k":
    print("Eritma ishqorli")
else: print("Eritma neytral")
```



4.8-masala. Agar kvadratning tomoni A, doiraning radiusi R ga teng bo‘lsa, kvadrat va doiraning yuzlarini solishtirib kattasini aniqlang.
Yechish. Kvadratning yuzi $s = a^2$, doiraning yuzi $k = \pi r^2$ formula yordamida aniqlanadi.

```
import math
A=float(input('Kvadratning tomoni\nA='))
R=float(input('Doiraning radiusi\nR='))
S=pow(A,2)
C=math.pi*pow(R,2)
if S>C:
    print("Kvadratning yuzi katta")
else: print("Doiraning yuzi katta")
```

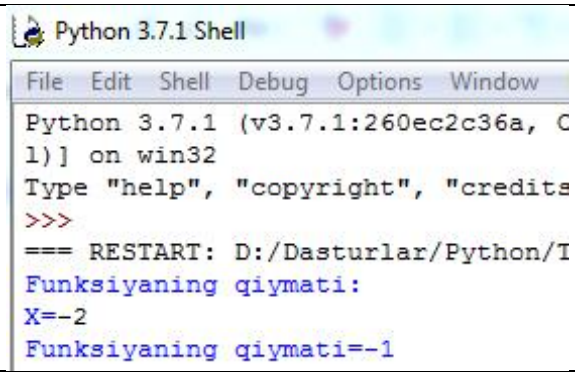


4.9-masala. Quyidagi funksiyani hisoblang: x>0 bo‘lganda 1 ga teng; x=0 da nolga teng; x<0 da -1 ga teng.

Yechish. Berilgan funksiya $y = sign(x)$ bilan belgilanadi.

$$signx = \begin{cases} 1, \ agar & x > 0 \\ 0, \ agar & x = 0 \\ -1, \ agar & x < 0 \end{cases}$$

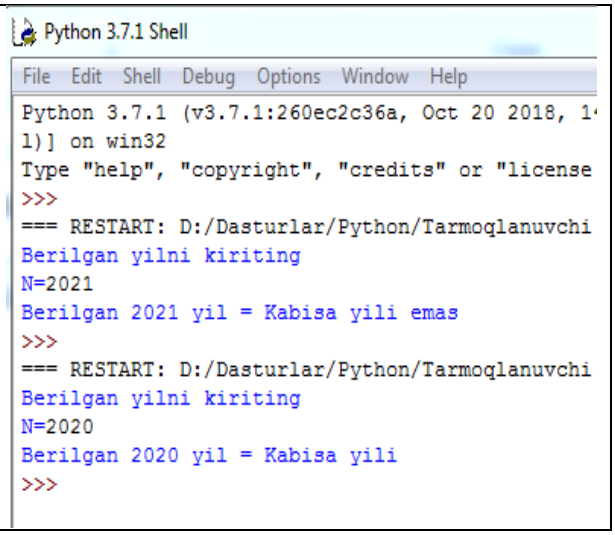
```
X=float(input('Funksiyaning qiymati:\nX='))
if X>0:
   print("Funksiyaning qiymati=1")
elif X==0:
   print("Funksiyaning qiymati=0")
else:
   print("Funksiyaning qiymati=-1")
```



4.10-masala. Berilgan N sonli yil kabisa yili bo‘lishi yoki bo‘lmasligini aniqlang. Agar N soni 100 ga karrali son bo‘lmasa va uning oxirgi ikki raqami 4 ga karrali son bo‘lsa, u holda N-yil kabisa yilidir. Agar N soni 100 karrali bo‘lsa,u holda N soni 400 ga karrali bo‘lgandagina mazkur yil kabisa yili bo‘ladi.

Yechish. Ushbu $w = n - floor(\frac{n}{u}) * u$ qoldiqni topish formulasini qism dasturga kiritib, undan n sonni u=100, u=400 va u=4 ga bo’lish natijasida hosil bo‘lgan qoldiqni topishda uch marta foydalanamiz.

```
N=int(input("Berilgan yilni kiriting\nN="))
Y=N%100
Z=N%10
if Y!=0 and Z%4==0:
   print("Berilgan",N,"yil = Kabisa yili")
else:
   print("Berilgan",N,"yil = Kabisa yili emas")
```

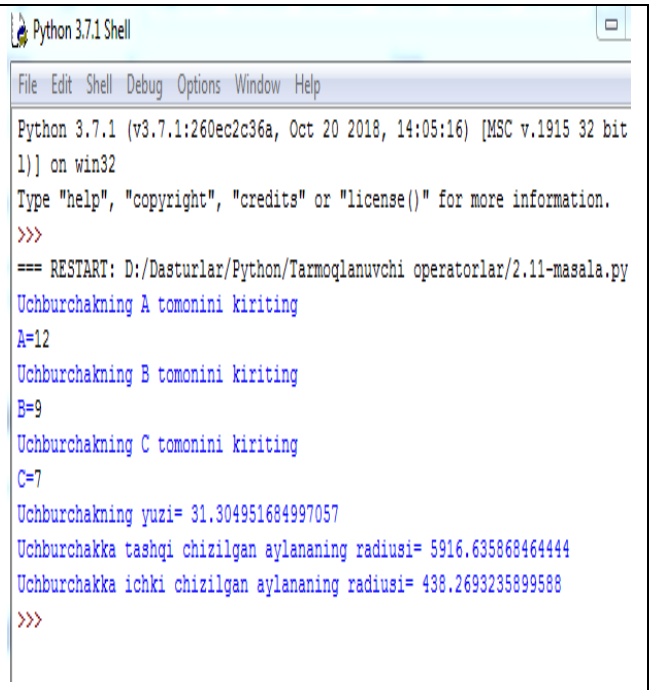


4.11-masala. A, B, C sonlar mos ravishda uchta kesmaning uzunliklarini ifodalaydi. Agar kesmalar uchburchakning tomonlarini ifodalasa, uchburchakning yuzi S, uchburchakka tashqi va ichki chizilgan aylanalarning radiuslari r1 va r2 larni toping.

Yechish. Agar $p = \frac{a+b+c}{2}$ deb belgilash kiritsak, uchburchakning mavjud bo’lish sharti $p \cdot (p-a) \cdot (p-b) \cdot (p-c) > 0$ ko‘rinishda yoziladi. Uchburchakning yuzi

$s=\sqrt{p\cdot(p-a)\cdot(p-b)\cdot(p-c)}$, tashqi aylananing radiusi $r_1=\frac{a\cdot b\cdot c}{4\cdot s}$, ichki aylananing radiusi esa $r_2=\frac{s}{p}$ formula yordamida aniqlanadi.

```
import math
A=float(input("Uchburchakning A tomonini kiriting\nA="))
B=float(input("Uchburchakning B tomonini kiriting\nB="))
C=float(input("Uchburchakning C tomonini kiriting\nC="))
if (A+B)>C and (A+C)>B and (B+C)>A:
    p=(A+B+C)/2
    S=math.sqrt(p*(p-A)*(p-B)*(p-C))
    r1=(A*B*C)/4*S
    r2=S*p
    print("Uchburchakning yuzi=",S)
    print("Uchburchakka tashqi chizilgan aylananing radiusi=",r1)
    print("Uchburchakka ichki chizilgan aylananing radiusi=",r2)
else:
    print("Berilgan sonlar bilan uchburchak yasab bo'lmaydi!")
```



4.12-masala. Ax+B=0 tenglamani yeching.

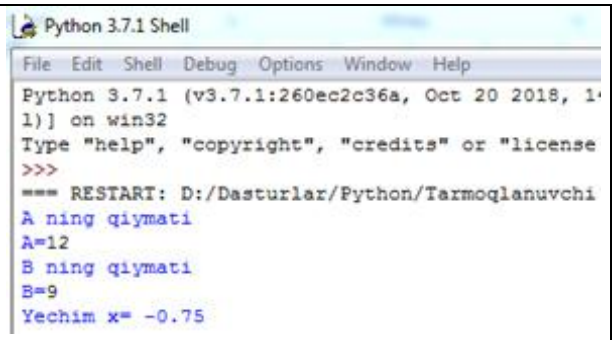
Yechish. Ma'lumki, $a\cdot x+b=0$ tenglamaning yechimi quyidagicha aniqlanadi:

1). A=0, b=0 bo'lsa, tenglama cheksiz ko'p yechimga ega;

2). A=0, $b\neq 0$ bo'lsa, tenglama yechimga ega emas;

3). $A\neq 0$, bo'lsa, tenglama $x=-\frac{b}{a}$ yagona yechimga ega;

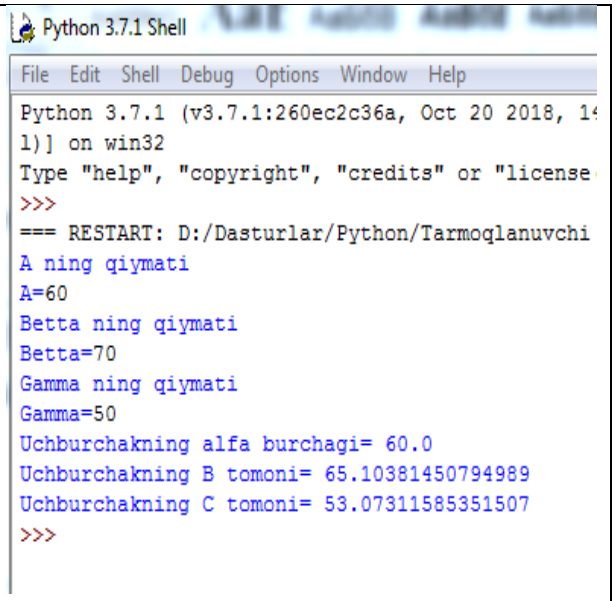
```
A=float(input('A ning qiymati\nA='))
B=float(input('B ning qiymati\nB='))
if A==0 and B==0:
    print("Tenglama cheksiz ko'p yechimga ega")
elif A==0 and B!=0:
    print("Tenglama yechimga ega emas")
else:
    x=-B/A
    print("Yechim x=",x)
```



4.13-masala. Bir tomoni va unga yopishgan ikkita burchagi berilgan uchburchakning uchinchi burchagi va qolgan ikki tomonini aniqlang.

Yechish. Uchburchakning a tomoni va b_1, c_1 burchaklari gradus o‘lchovida berilgan.a_1 burchakni $a_1 = 180 - (b1 + c1)$ formula yordamida aniqlaymiz. a_1, b_1, c_1 burchaklarlarning radian o‘lchovidagi kattaligini a_2, b_2, c_2 bilan belgilasak, $a_2 = \frac{\pi \cdot a_1}{180}$; $b_2 = \frac{\pi \cdot b_1}{180}$; $c_2 = \frac{\pi \cdot c_1}{180}$; formulalar o’rinli bo‘ladi. Bunda $\pi = 3{,}14159$ b va c tomonlarni sinuslar teoremasiga asosan aniqlaymiz: $b = \frac{a \cdot \sin b_2}{\sin a_2}$; $c = \frac{a \cdot \sin c_2}{\sin a_2}$;

```
import math
A=float(input("A ning qiymati\nA="))
Betta=float(input("Betta ning
qiymati\nBetta="))
Gamma=float(input("Gamma ning
qiymati\nGamma="))
alfa=180-(Betta+Gamma)
alfa2=(math.pi*alfa)/180
Betta2=(math.pi*Betta)/180
Gamma2=(math.pi*Gamma)/180
B=(A*math.sin(Betta2))/math.sin(alfa2)
C=A*math.sin(Gamma2)/math.sin(alfa2)
print("Uchburchakning alfa burchagi=",alfa)
print("Uchburchakning B tomoni=",B)
print("Uchburchakning C tomoni=",C)
```



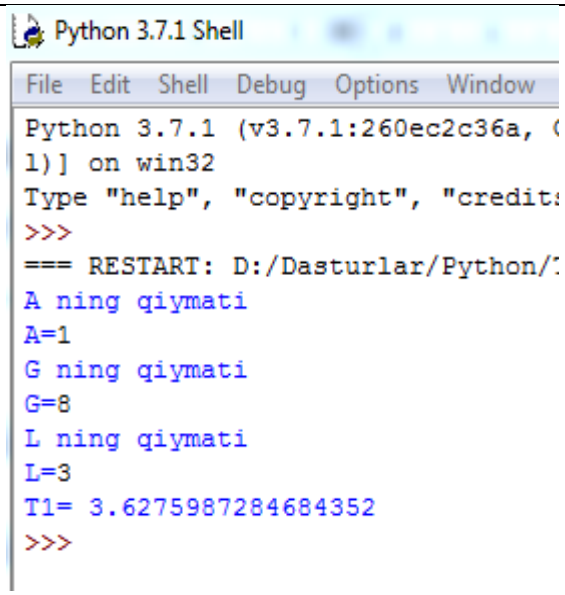
4.14-masala. Uchta sonning berilgan, ularninig eng kattasi (EKT) ni toping.

```
x=int(input("Birinchi son\nx="))
y=int(input("Ikkinchi son\ny="))
z=int(input("Uchinchi son\nz="))
if x>y:
   max=x
else: max=y
if max>z:
   max=max
else: max=z
print("Eng katta son max=",max)
```

4.15-masala. Uzunligi 1 ga teng matematik mayatnikning osilgan nuqtasi qo‘zg‘almas yoki yuqoriga yoki pastga tezlanish bilan harakatlangan hollarda uning tebranish davri aniqlansin.

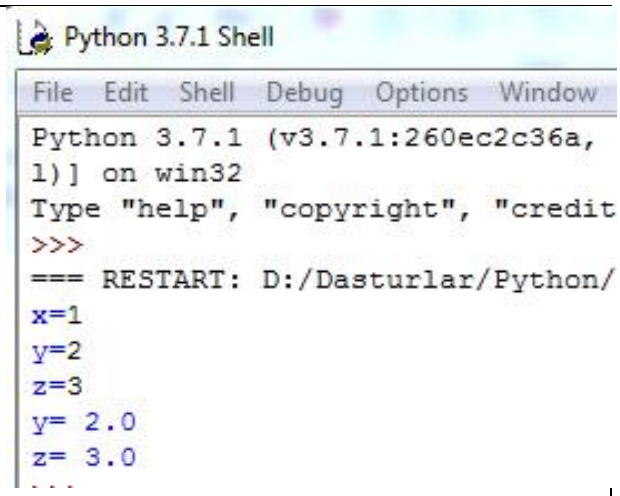
Yechish. Agar mayatnik osilgan nuqta qo‘zg‘almas bo‘lsa, $T = 2 \cdot \pi \cdot \sqrt{\frac{l}{g}}$; mayatnik osilgan nuqta yuqoriga a tezlanish bilan harakatlansa, $T_1 = 2 \cdot \pi \cdot \sqrt{\frac{l}{g+a}}$; mayatnik osilgan nuqta pastga a tezlanish bilan harakatlansa, $T_2 = 2 \cdot \pi \cdot \sqrt{\frac{l}{a-g}}$; formulalar o‘rinli bo‘ladi. Bunda $\pi = 3{,}14159$, g=9,81 deb olish mumkin. Agar a=g bo‘lsa, mayatnik vaznsizlik holatida bo‘ladi va bu holatda mayatnik tebranmaydi.

```
import math
A=float(input("A ning qiymati\nA="))
G=float(input("G ning qiymati\nG="))
L=float(input("L ning qiymati\nL="))
if A==0:
    T=2*math.pi*math.sqrt(L/G)
    print("T=",T)
elif A==G:
    print("Mayatnik vaznsiz holatda bo'ladi")
elif A<G:
    T1=2*math.pi*math.sqrt(L/(G+A))
    print("T1=",T1)
else:
    T2=2*math.pi*math.sqrt(L/(G-A))
    print("T2=",T2)
```



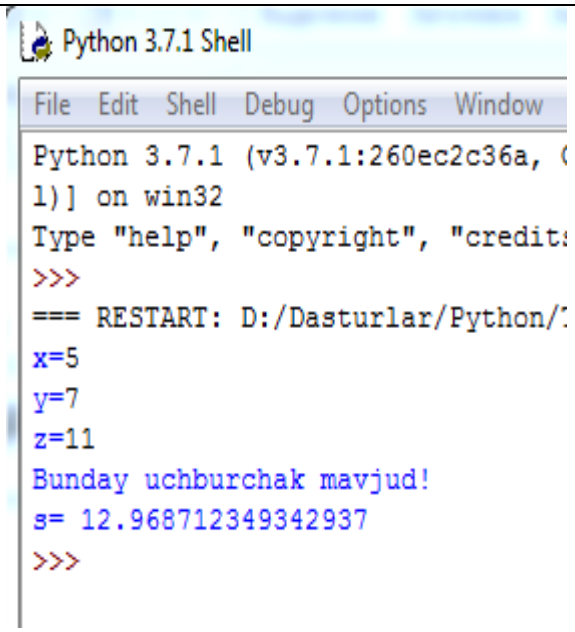
4.16-masala. Uchta X, Y, Z haqiqiy sonlar berilgan. Bu sonlardan qaysi biri (1,5) intervalga tegishli ekanligini aniqlang.
Yechish. (1,5) intervalga tegishli sonlarni aniqlashni qism-dastur yordamida kiritamiz.

```
x=float(input("x="))
y=float(input("y="))
z=float(input("z="))
if x>1 and x<5:
    print("x=",x)
if y>1 and y<5:
    print("y=",y)
if z>1 and z<5:
    print("z=",z)
```



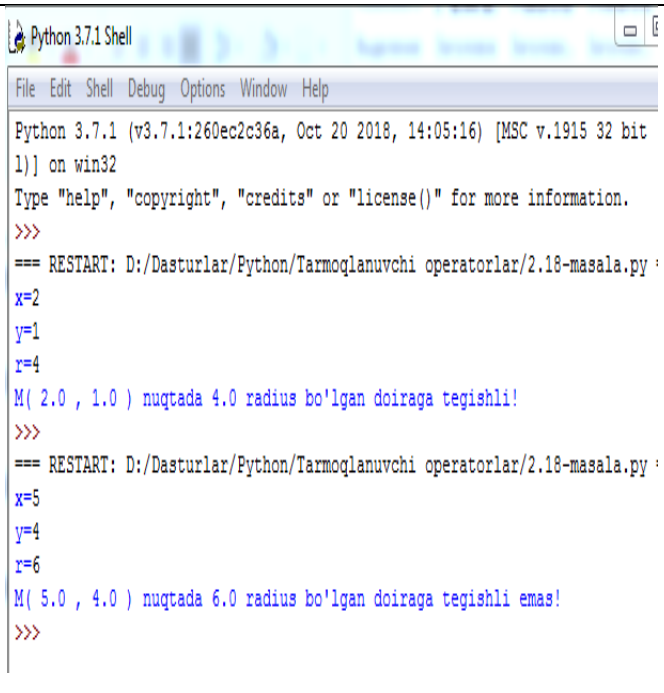
4.17-masala. Uchta X, Y, Z musbat sonlar berilgan. Tomonlari X, Y, Z ga teng uchburchak mavjudmi? Agar mavjud bo‘lsa bu uchburchakning yuzini toping.

```
import math
x=int(input("x="))
y=int(input("y="))
z=int(input("z="))
if (x+y)>z and (x+z)>y and (z+y)>x:
    print("Bunday uchburchak mavjud!")
    p=(x+y+z)/2
    s=math.sqrt(p*(p-x)*(p-y)*(p-z))
    print("s=",s)
else:
    print("Bunday uchburchak mavjud emas!")
```



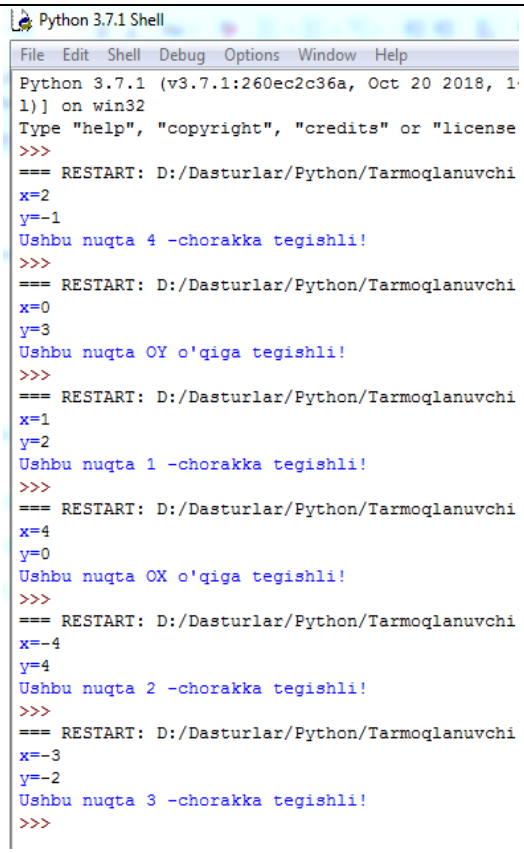
4.18-masala. Koordinatalari berilgan M(X,Y) nuqtaning radiusi R ga teng va markazi koordinatalar boshida bo‘lgan doiraga tegishli bo‘lishini aniqlang.

```
import math
x=float(input('x='))
y=float(input('y='))
r=float(input('r='))
if
(math.pow(x,2)+math.pow(y,2))>math
.pow(r,2):
    print('M(',x,',',y,') nuqtada',r,"radius
bo'lgan doiraga tegishli emas!")
else:
    print('M(',x,',',y,') nuqtada',r,"radius
bo'lgan doiraga tegishli!")
```



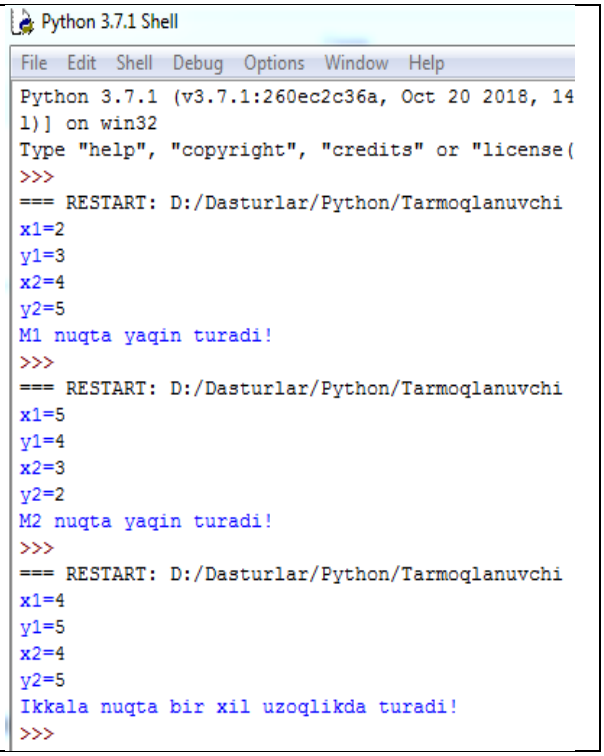
4.19-masala. Koordinatalari berilgan M(X,Y) nuqtaning koordinata tekisligining qaysi choragida ekanligini aniqlaydigan dastur tuzing.

```
x=int(input('x='))
y=int(input('y='))
if x<0 and y<0:
    k=3
if (x<0 or x>0) and y==0:
    k=0
if x<0 and y>0:
    k=2
if x>0 and y<0:
    k=4
if x>0 and y>0:
    k=1
if x==0 and (y<0 or y>0):
    k=5
if k==0:
    print("Ushbu nuqta OX o'qiga tegishli!")
elif k==5:
    print("Ushbu nuqta OY o'qiga tegishli!")
else:
    print("Ushbu nuqta",k,"-chorakka tegishli!")
```



4.20-masala. Koordinatalari berilgan ikkita M1(X1,Y1) va M2(X2,Y2) nuqtalarning qaysi biri koordinata boshiga yaqin turishini aniqlash dasturi.

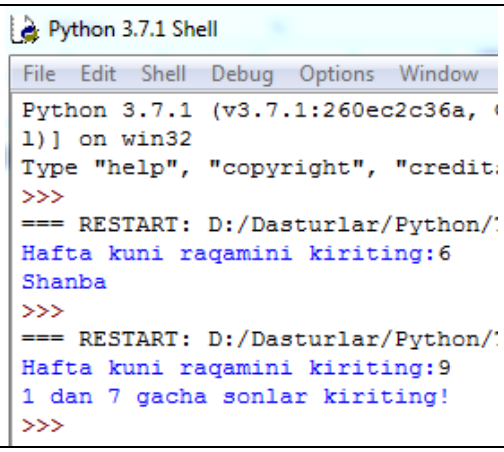
```
import math
x1=int(input('x1='))
y1=int(input('y1='))
x2=int(input('x2='))
y2=int(input('y2='))
r1=math.sqrt(math.pow(x1,2)+math.pow(y1,2)
)
r2=math.sqrt(math.pow(x2,2)+math.pow(y2,2)
)
if r1>r2:
    print("M2 nuqta yaqin turadi!")
elif r1<r2:
    print("M1 nuqta yaqin turadi!")
else:
    print("Ikkala nuqta bir xil uzoqlikda
turadi!")
```



3.4. PYTHON DA SHARTLI TANLASH ALGORITMLARIGA DASTUR TUZISH

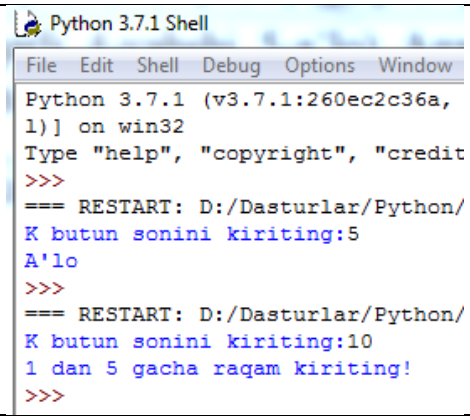
5.1-masala. 1-7 gacha bo‘lgan butun sonlar berilgan. Kiritilgan songa mos ravishda hafta kunlarini so‘zda ifodalovchi dastur tuzing. (1-Dushanba.2-Chorshanba....h.k)

```
k=int(input('Hafta kuni raqamini kiriting:'));
switch={
   1: 'Dushanba',
   2: 'Seshanba',
   3: 'Chorshanba',
   4: "Payshanba",
   5: 'Juma',
   6: 'Shanba',
   7: 'Yakshanba',
   }
print(switch.get(k,"1 dan 7 gacha sonlar kiriting!"));
```



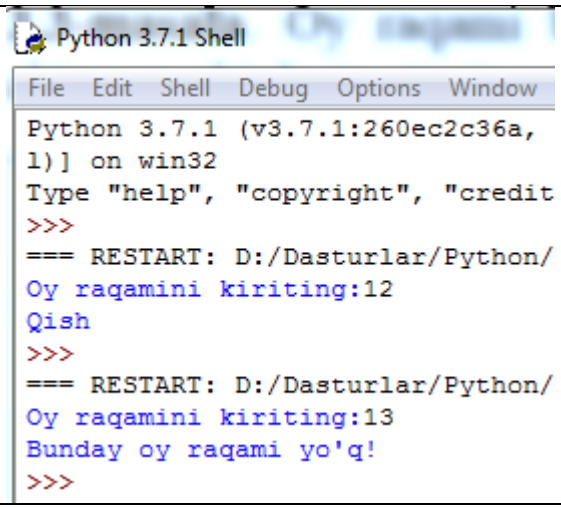
5.2-masala. K butun soni berilgan. Baho natijalarini chiqaruvchi dasturini tuzing.(1-yomon, 2-qoniqarsiz, 3- qoniqarli, 4-yahshi, 5-a’lo). Agar к soni 1-5 gacha oraliqqa tegishli bo‘lmasa, u holda “xato 1 dan 5 gacha raqam kiriting!”” matni chiqarilsin.

```
k=int(input('K butun sonini kiriting:'));
switch={
   1: 'Yomon',
   2: 'Qoniqarsiz',
   3: 'Qoniqarli',
   4: "Yaxshi",
   5: "A'lo"
   }
print(switch.get(k,"1 dan 5 gacha raqam kiriting!"));
```



5.3-masala. Oy raqami berilgan. Kiritilgan oy qaysi faslga tegishli ekanligini chiqaruvchi dastur tuzing. (Masalan: 2 chi oy, “qish")

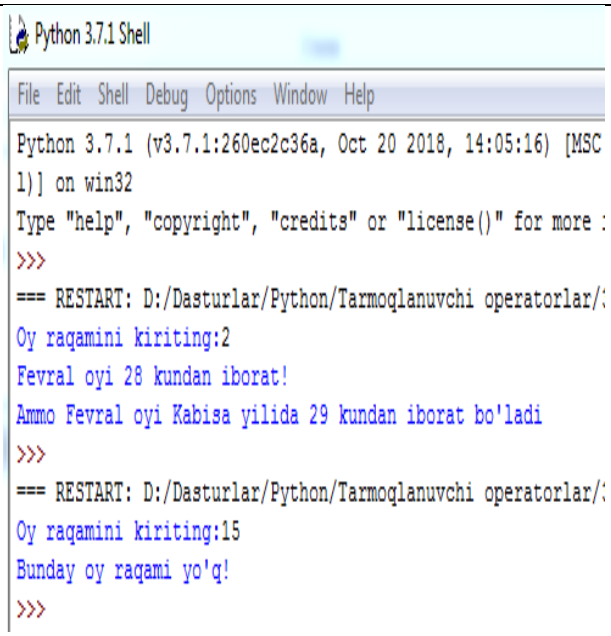
```
son=int(input('Oy raqamini kiriting:'));
switch={
   1: 'Qish',
   2: 'Qish',
   3: 'Bahor',
   4: "Bahor",
   5: 'Bahor',
   6: 'Yoz',
   7: 'Yoz',
   8: 'Yoz',
   9: "Kuz",
```



```
    10: 'Kuz',
    11: 'Kuz',
    12: 'Qish'
    }
print(switch.get(son,"Bunday oy raqami
yo'q!"));
```

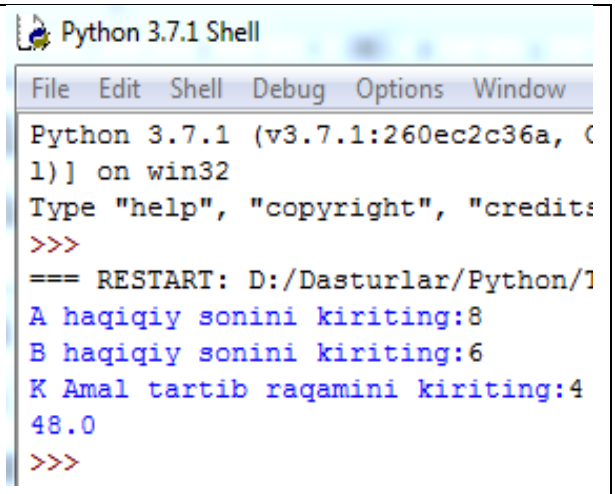
5.4-masala. Oy raqami berilgan. Shu oyda nechta kun borligini aniqlovchi dastur tuzing.

```
son=int(input('Oy raqamini kiriting:'));
switch={
    1: 'Yanvar oyi 31 kundan iborat!',
    2: "Fevral oyi 28 kundan iborat!\nAmmo
Fevral oyi Kabisa yilida 29 kundan iborat
bo'ladi",
    3: 'Mart oyi 31 kundan iborat!',
    4: "Aprel oyi 30 kundan iborat!",
    5: 'May oyi 31 kundan iborat!',
    6: 'Iyun oyi 30 kundan iborat!',
    7: 'Iyul oyi 31 kundan iborat!',
    8: 'Avgust oyi 31 kundan iborat!',
    9: "Sentabr oyi 30 kundan iborat!",
    10: 'Oktabr oyi 31 kundan iborat!',
    11: 'Noyabr oyi 30 kundan iborat!',
    12: 'Dekabr oyi 31 kundan iborat!'
    }
print(switch.get(son,"Bunday oy raqami yo'q!"));
```



5.5-masala. A, B haqiqiy butun soni va K-amal tartib raqami berilgan. A va B sonlari ustida arifmetik amallar bajaruvchi dastur tuzing. K-amal quyidagi qiymatlarni qabul qiladi: 1-qo‘shish, 2-ayirish, 3-bo‘lish, 4- ko‘paytirish.

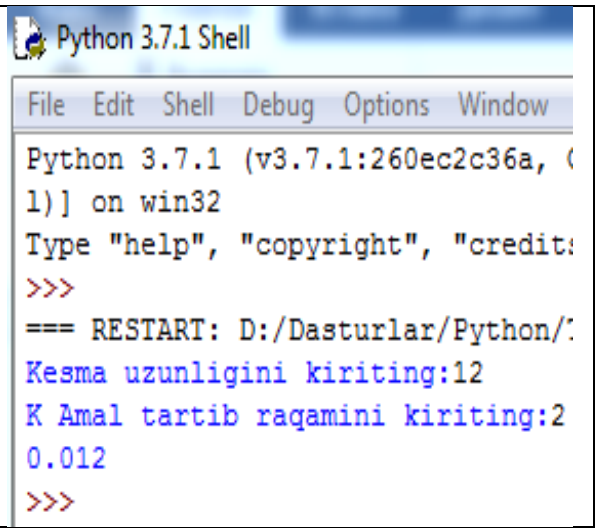
```
A=float(input('A haqiqiy sonini kiriting:'));
B=float(input('B haqiqiy sonini kiriting:'));
k=int(input('K Amal tartib raqamini
kiriting:'));
switch={
    1: A+B,
    2: A-B,
    3: A/B,
    4: A*B
    }
print(switch.get(k,"1 dan 4 gacha raqam
kiriting!"));
```



5.6-masala Uzinlik birliklari quyidagi tartibda berilgan. 1-detsimetr, 2-kilometr, 3-

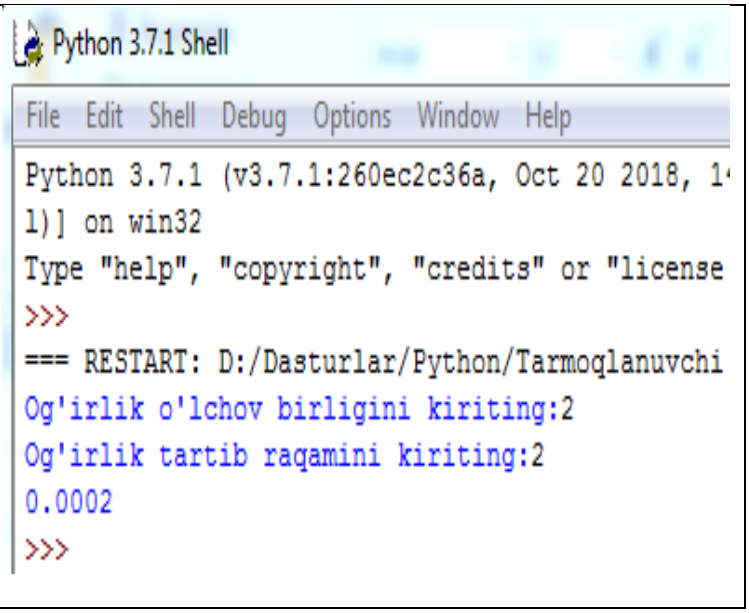
metr, 4-millimeter, 5- santimetr. Uzunlik birligini bildiruvchi son berilgan (1 - 5 oraliqda) va shu birlikdagi kesma uzunligi berilgan (haqiqiy son). Kesmaning uzunligini metrlarda ifodalovchi dastur tuzing.

```
a=float(input('Kesma uzunligini kiriting:'));
k=int(input('K Amal tartib raqamini kiriting:'));
switch={
    1: a/10,
    2: a/1000,
    3: a,
    4: a/1000,
    5: a/100
    }
print(switch.get(k,"1 dan 5 gacha raqam kiriting!"));
```



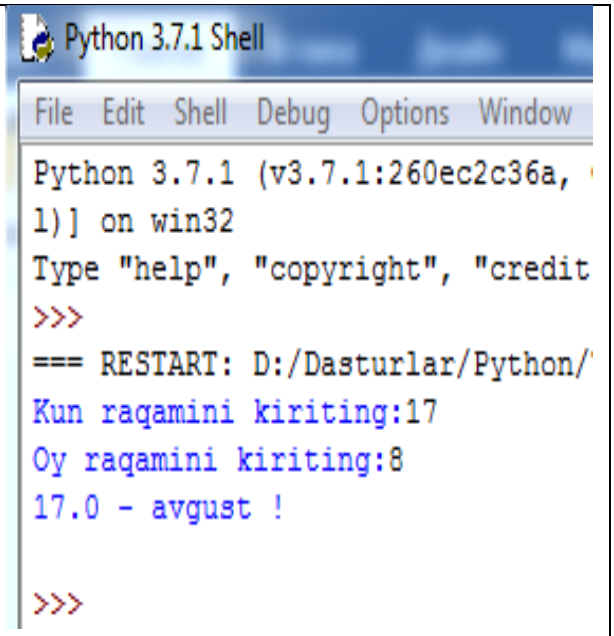
5.7-masala. Og‘irlik birliklari quyidagi tartibda berilgan. 1-kilogramm, 2-milligramm, 3-gramm, 4-tonna, 5- sentner. Og‘irlik birligini bildiruvchi son berilgan va shu birlikdagi og‘irlik qiymati berilgan. Og‘irlikni kilogramda ifodalovchi dastur tuzing.

```
a=float(input("Og'irlik o'lchov birligini kiriting:"));
k=int(input("Og'irlik tartib raqamini kiriting:"));
switch={
    1: a,
    2: a/10000,
    3: a/1000,
    4: a*1000,
    5: a*100
    }
print(switch.get(k,"Og'irlik tartib raqamini 1 dan 5 gacha kiriting!"));
```



5.8-masala. Sanani bildiruvchi ikkita butun son berilgan D (kun) va M (oy). (Kabisa bo‘lmagan yil sanasi kiritiladi). Berilgan sanani ifodalovchi dastur tuzing. Kabisa yilida 366 kun, kabisa bo‘lmagan yilda 365 kun mavjud.

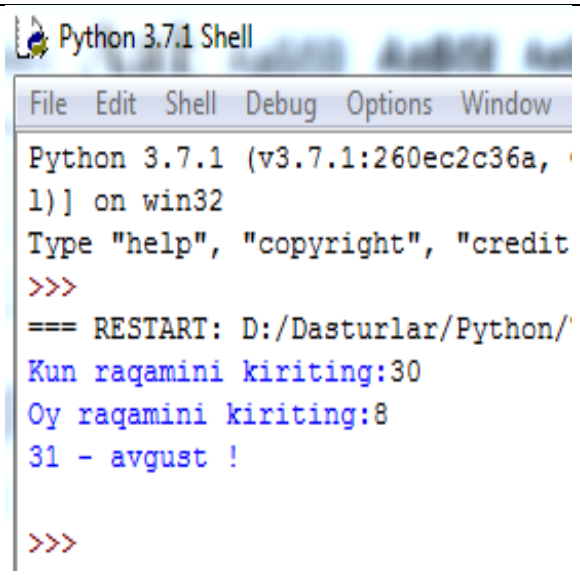
```
d=float(input("Kun raqamini kiriting:"));
m=int(input("Oy raqamini kiriting:"));
if m==1:
    if d>31:
        print("Yanvar oyida bunday sana
yo'q!");
    else:
        m='yanvar';
        print(d,'-',m,'!\n');
elif m==2:
    if d>28:
        print("Fevral oyida bunday sana
yo'q!");
    else:
        m='fevral';
        print(d,'-',m,'!\n');
elif m==3:
    if d>31:
        print("Mart oyida bunday sana yo'q!");
    else:
        m='mart';
        print(d,'-',m,'!\n');
elif m==4:
    if d>30:
        print("Aprel oyida bunday sana
yo'q!");
    else:
        m='aprel';
        print(d,'-',m,'!\n');
elif m==5:
    if d>31:
        print("May oyida bunday sana yo'q!");
    else:
        m='may';
        print(d,'-',m,'!\n');
elif m==6:
    if d>30:
        print("Iyun oyida bunday sana yo'q!");
    else:
        m='iyun';
        print(d,'-',m,'!\n');
elif m==7:
    if d>31:
        print("Iyul oyida bunday sana yo'q!");
    else:
```



```
        m='iyul';
        print(d,'-',m,'!\n');
elif m==8:
    if d>31:
        print("Avgust oyida bunday sana
yo'q!");
    else:
        m='avgust';
        print(d,'-',m,'!\n');
elif m==9:
    if d>30:
        print("Sentabr oyida bunday sana
yo'q!");
    else:
        m='sentabr';
        print(d,'-',m,'!\n');
elif m==10:
    if d>31:
        print("Oktabr oyida bunday sana
yo'q!");
    else:
        m='oktabr';
        print(d,'-',m,'!\n');
elif m==11:
    if d>30:
        print("Noyabr oyida bunday sana
yo'q!");
    else:
        m='noyabr';
        print(d,'-',m,'!\n');
elif m==12:
    if d>31:
        print("Dekabr oyida bunday sana
yo'q!");
    else:
        m='dekabr';
        print(d,'-',m,'!\n');
else:
    print("Bunday oy raqami yo'q!");
```

5.9-masala. Ikkita butun son berilgan D (kun) va M (oy). (Kabisa bo‘lmagan yil sanasi kiritiladi). Berilgan sanadan keyingi sanani ifodalovchi dastur tuzing.

```
d=int(input("Kun raqamini kiriting:"));
m=int(input("Oy raqamini kiriting:"));
if m==1:
    d=d+1;
    if d>31:
        print("Yanvar oyida bunday sana
yo'q!");
    else:
        m='yanvar';
        print(d,'-',m,'!\n');
elif m==2:
    d=d+1;
    if d>28:
        print("Fevral oyida bunday sana yo'q!");
    else:
        m='fevral';
        print(d,'-',m,'!\n');
elif m==3:
    d=d+1;
    if d>31:
        print("Mart oyida bunday sana yo'q!");
    else:
        m='mart';
        print(d,'-',m,'!\n');
elif m==4:
    d=d+1;
    if d>30:
        print("Aprel oyida bunday sana yo'q!");
    else:
        m='aprel';
        print(d,'-',m,'!\n');
elif m==5:
    d=d+1;
    if d>31:
        print("May oyida bunday sana yo'q!");
    else:
        m='may';
        print(d,'-',m,'!\n');
elif m==6:
    d=d+1;
    if d>30:
        print("Iyun oyida bunday sana yo'q!");
    else:
        m='iyun';
        print(d,'-',m,'!\n');
```

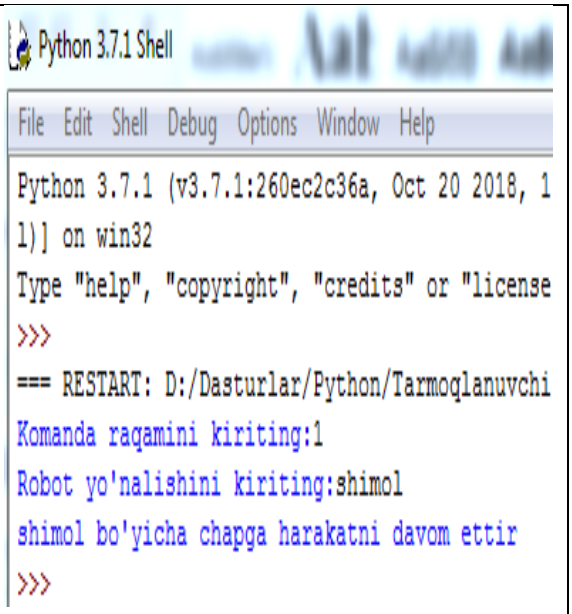


```
elif m==7:
   d=d+1;
   if d>31:
      print("Iyul oyida bunday sana yo'q!");
   else:
      m='iyul';
      print(d,'-',m,'!\n');
elif m==8:
   d=d+1;
   if d>31:
      print("Avgust oyida bunday sana
yo'q!");
   else:
      m='avgust';
      print(d,'-',m,'!\n');
elif m==9:
   d=d+1;
   if d>30:
      print("Sentabr oyida bunday sana
yo'q!");
   else:
      m='sentabr';
      print(d,'-',m,'!\n');
elif m==10:
   d=d+1;
   if d>31:
      print("Oktabr oyida bunday sana
yo'q!");
   else:
      m='oktabr';
      print(d,'-',m,'!\n');
elif m==11:
   d=d+1;
   if d>30:
      print("Noyabr oyida bunday sana
yo'q!");
   else:
      m='noyabr';
      print(d,'-',m,'!\n');
elif m==12:
   d=d+1;
   if d>31:
      print("Dekabr oyida bunday sana
yo'q!");
   else:
```

```
        m='dekabr';
        print(d,'-',m,'!\n');
else:
   print("Bunday oy raqami yo'q!");
```

5.10-masala. Robot faqat to‘rtta tomonga ko‘cha oladi(‘v’-shimol, ‘j’-janub, ‘q’-sharq, ‘g’-g‘arb) va uchta raqamli buyruq: 0-harakni davom ettir, 1-chapga buril, 2-o‘ngga buril. Y - robot yo‘nalishi va K - buyruq berilgan. Berilgan buyruq bajarilgandan keying robot holatini aniqlovchi dastur tuzing.

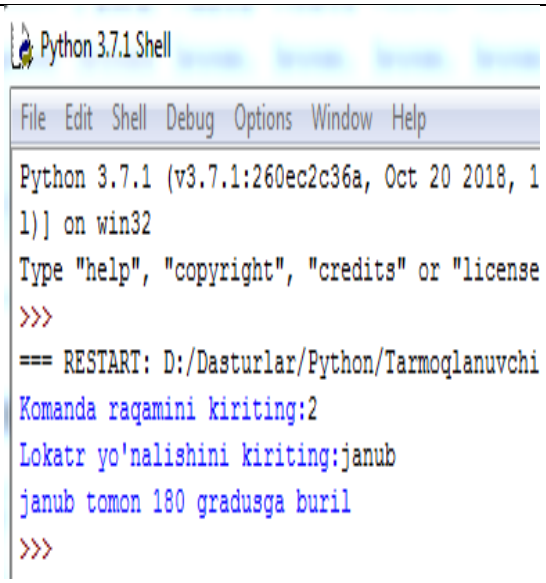
```
k=int(input("Komanda raqamini
kiriting:"));
y=input("Robot yo'nalishini kiriting:");
s="shimol";j="janub";q="sharq";g="g'arb";
if k==0:
   if y==s:
      print(y,"bo'yicha harakatni davom
ettir");
   elif y==j:
      print(y,"bo'yicha harakatni davom
ettir");
   elif y==q:
      print(y,"bo'yicha harakatni davom
ettir");
   elif y==g:
      print(y,"bo'yicha harakatni davom
ettir");
   else:
      print("Yo'nalish yoki komandani
to'g'ri kiriting!");
elif k==1:
   if y==s:
      print(y,"bo'yicha chapga harakatni
davom ettir");
   elif y==j:
      print(y,"bo'yicha chapga harakatni
davom ettir");
   elif y==q:
      print(y,"bo'yicha chapga harakatni
davom ettir");
   elif y==g:
      print(y,"bo'yicha chapga harakatni
davom ettir");
   else:
      print("Yo'nalish yoki komandani
```



```
to'g'ri kiriting!");
elif k==2:
    if y==s:
        print(y,"bo'yicha o'ngga harakatni
davom ettir");
    elif y==j:
        print(y,"bo'yicha o'ngga harakatni
davom ettir");
    elif y==q:
        print(y,"bo'yicha o'ngga harakatni
davom ettir");
    elif y==g:
        print(y,"bo'yicha o'ngga harakatni
davom ettir");
    else:
        print("Yo'nalish yoki komandani
to'g'ri kiriting!");
else:
    print("Bunday yo'nalish yoki komanda
mavjud emas!");
```

5.11-masala. Lokator dunyoning bir tomoniga qaratilgan((‘v’-shimol, ‘j’-janub, ‘q’-sharq, ‘g’-g‘arb) va uchta raqamli buyruq: 0-o‘ngga buril, 1-chapga buril, 2-burilish 180°. C - lokatorning boshlang‘ich holati va K1, K2 - buyruqlar berilgan. Berilgan buyruq bajarilgandan keyin lokator holatini aniqlovchi dastur tuzing.

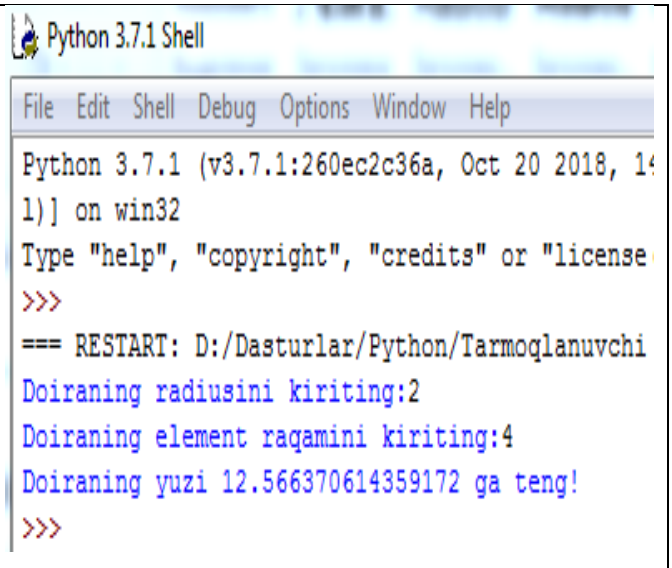
```
k=int(input("Komanda raqamini
kiriting:"));
y=input("Lokatr yo'nalishini kiriting:");
s="shimol";j="janub";q="sharq";g="g'arb";
if k==0:
    if y==s:
        print(y,"bo'yicha o'ngga buril");
    elif y==j:
        print(y,"bo'yicha o'ngga buril");
    elif y==q:
        print(y,"bo'yicha o'ngga buril");
    elif y==g:
        print(y,"bo'yicha o'ngga buril");
    else:
        print("Yo'nalish yoki komandani
to'g'ri kiriting!");
elif k==1:
    if y==s:
        print(y,"bo'yicha chapga buril");
```



```
    elif y==j:
        print(y,"bo'yicha chapga buril");
    elif y==q:
        print(y,"bo'yicha chapga buril");
    elif y==g:
        print(y,"bo'yicha chapga buril");
    else:
        print("Yo'nalish yoki komandani
to'g'ri kiriting!");
elif k==2:
    if y==s:
        print(y,"tomon 180 gradusga buril");
    elif y==j:
        print(y,"tomon 180 gradusga buril");
    elif y==q:
        print(y,"tomon 180 gradusga buril");
    elif y==g:
        print(y,"tomon 180 gradusga buril");
    else:
        print("Yo'nalish yoki komandani
to'g'ri kiriting!");
else:
    print("Bunday yo'nalish yoki komanda
mavjud emas!");
```

5.12-masala. Doiraning elementlari quyidagi tartibda nomerlangan. 1-radius R, 2-diametr $D = 2 * R$, 3-uzunligi $L = 2 * \pi * R$, 4-doiraning yuzasi $S = \pi * R^2$. Shu formulalardan bittasi berilganda qolganlarini topuvchi dastur tuzing.

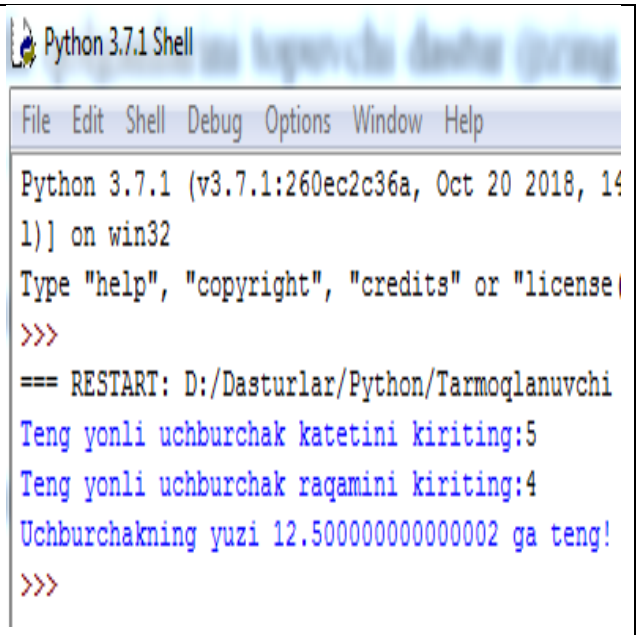
```
import math;
r=float(input('Doiraning radiusini
kiriting:'));
k=float(input('Doiraning element
raqamini kiriting:'));
if k==1:
    print("Radius",r,"ga teng!");
elif k==2:
    D=2*r;
    print("Diametr",D,"ga teng!");
elif k==3:
    L=2*math.pi*r;
    print("Aylana uzunligi",L,"ga teng!");
elif k==4:
    S=math.pi*pow(r,2);
    print("Doiraning yuzi",S,'ga teng!');
```



else: print("Doiraning element raqamini 1 dan 4 gacha kiriting!");	

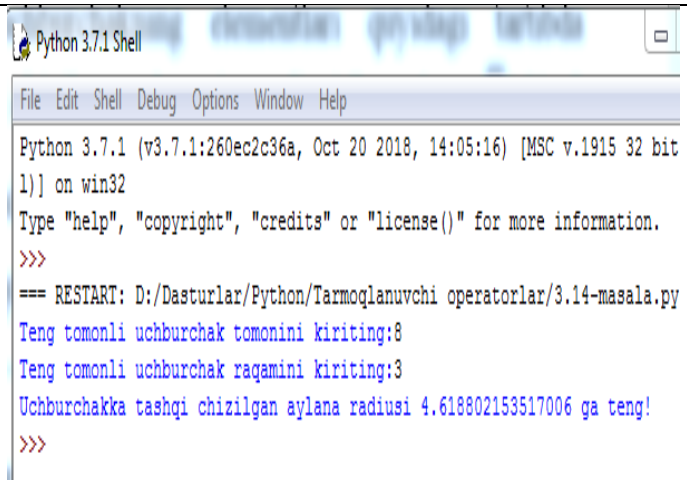
5.13-masala. Teng yonli uchburchakning elementlari quyidagi tartibda nomerlangan. 1-katet – “a”, 2-katet - “b”, 3-gipotenuza - “c” ($C = a * \sqrt{2}$) , 4- gipotenuzaga tushirilgan balandlik $h = c/2$, 5- yuzasi $S = (c * h)/2$. Ushbu formulalardan bittasi berilganda qolganlarini topuvchi dastur tuzing.

```
import math;
a=int(input('Teng yonli uchburchak katetini kiriting:'));
k=int(input('Teng yonli uchburchak raqamini kiriting:'));
if k==1:
   print('Kateti',a,'ga teng!');
elif k==2:
   c=a*math.sqrt(2);
   print('Gipotenuzasi',c,'ga teng!');
elif k==3:
   c=a*math.sqrt(2);
   h=c/2;
   print('Gipotenuzaga tushirilgan balandlik',h,'ga teng!');
elif k==4:
   c=a*math.sqrt(2);
   h=c/2;
   S=(c*h)/2;
   print("Uchburchakning yuzi",S,'ga teng!');
else:
   print('Teng yonli uchburchak raqamini 1 dan 4 gacha kiriting!');
```



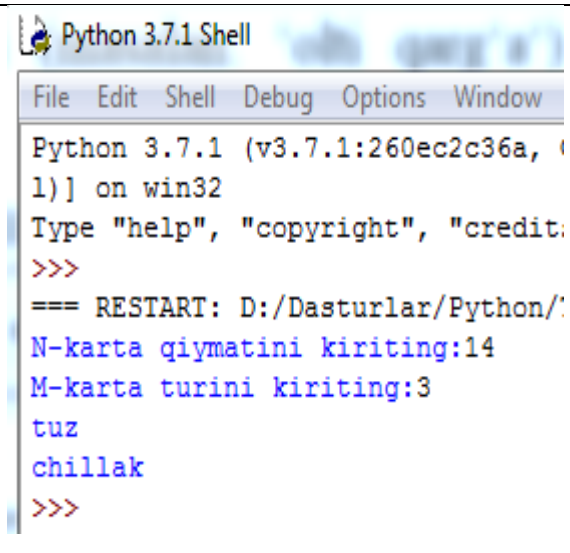
5.14-masala. Teng tomonli uchburchakning elementlari quyidagi tartibda nomerlangan. 1-tomoni a, 2-ichki chizilgan aylananing radiusi $R_1 = (a * \sqrt{3})/6$, 3 - tashqi chizilgan aylananing radiusi $R_2 = 2 * R_1$, 4-yuzasi $S = (a^2 * \frac{\sqrt{3}}{4})$. Shu formulalardan bittasi berilganda qolganlarini topuvchi dastur tuzing.

```
import math;
a=int(input('Teng tomonli uchburchak tomonini kiriting:'));
k=int(input('Teng tomonli uchburchak raqamini kiriting:'));
if k==1:
    print('Tomoni',a,'ga teng!');
elif k==2:
    R1=(a*math.sqrt(3))/6;
    print('Uchburchakka ichki chizilgan aylana radiusi',R1,'ga teng!');
elif k==3:
    R1=(a*math.sqrt(3))/6;
    R2=2*R1;
    print('Uchburchakka tashqi chizilgan aylana radiusi',R2,'ga teng!');
elif k==4:
    S=(a*a*math.sqrt(3))/4;
    print('Uchburchakning yuzi',S,'ga teng!');
else:
    print('Teng tomonli uchburchak raqamini 1 dan 4 gacha kiriting!');
```



5.15-masala. O‘yin kartasi turlari berilgan 1-gisht, 2-olma. 3-chillak, 4-qarg‘a. 10 lik kartadan katta kartalar quyidagi qiymatlarni o‘zlashtirgan: 11-valet, 12-dama, 13-qirol, 14-tuz. Ikkita butun son berilgan N-karta qiymati($6 \leq N \leq 14$), M-karta turi ($1 \leq M \leq 14$) kiritilganda karta nomlarini (masalan: ‘olti qarg‘a‘) chiqarib beruvchi dastur tuzing.

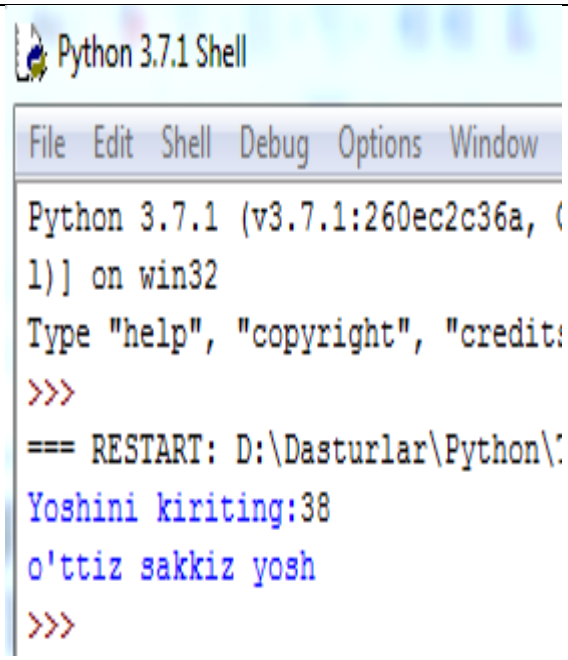
```
N=int(input('N-karta qiymatini kiriting:'));
M=int(input('M-karta turini kiriting:'));
switch={
    6: 'olti',
    7: 'yetti',
    8: 'sakkiz',
    9: "to'qqiz",
    10: "o'n",
    11: 'valet',
    12: 'dama',
    13: 'qirol',
    14: 'tuz'
    }
```



```
print(switch.get(N,"Karta qiymatini
6<=N<=14 oraliqda kiriting!"));
switch={
    1: "g'isht",
    2: 'olma',
    3: 'chillak',
    4: "qarg'a"
    }
print(switch.get(M,"Karta turining qiymatini
birdan to'rtgacha qiymatda kiriting!"));
```

5.16-masala. Yoshni yillarda aniqlovchi 20-69 gacha butun son berilgan. Son kiritilganda unga mos so'zlarda ifodalovchi dastur tuzing. (“yigirma yosh”, “qirq uch yosh” va h.k.)

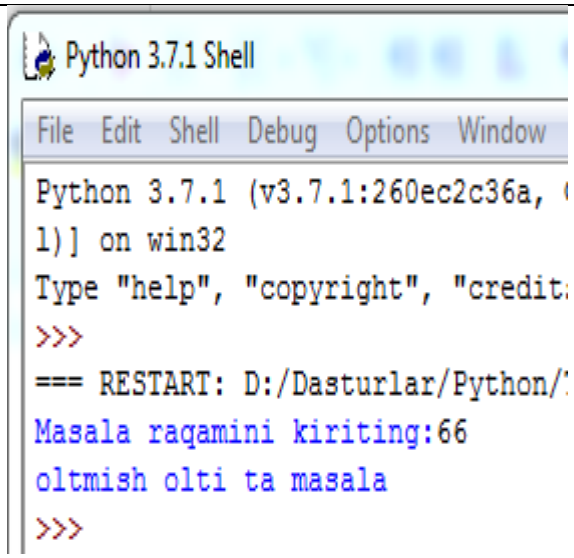
```
y=int(input('Yoshini kiriting:'));
n=int(y/10);
switch={
    1: "o'n",
    2: "yigirma",
    3: "o'ttiz",
    4: "qirq",
    5: "ellik",
    6: "oltmish",
    7: "yetmish",
    8: "sakson",
    9: "to'qson"
    }
m=y%10;
if m==1:
    bir="bir";
elif m==2:
    bir="ikki";
elif m==3:
    bir="uch";
elif m==4:
    bir="to'rt";
elif m==5:
    bir="besh";
elif m==6:
    bir="olti";
elif m==7:
    bir="yetti";
elif m==8:
    bir="sakkiz";
elif m==9:
```



```
    bir="to'qqiz";
print(switch.get(n,""),bir,'yosh');
```

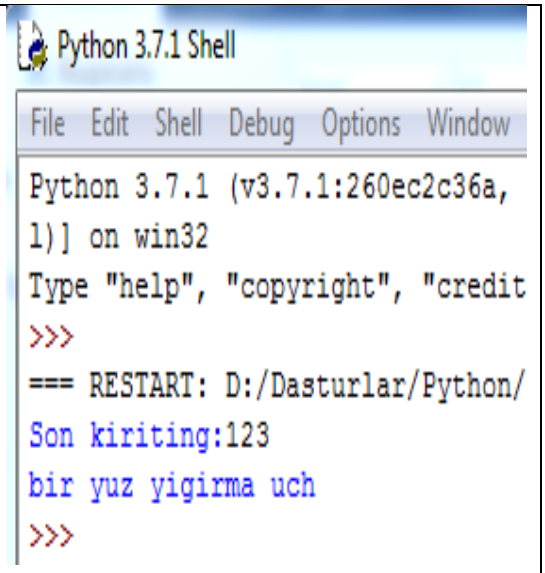
5.17-masala. O‘quv masalalarini aniqlovchi 1040 gacha butun son berilgan. Son kiritilganda unga mos so‘zlarda ifodalovchi dastur tuzing. (“yigirmata masala”, “o‘n uchta masala” va h.k.)

```
y=int(input('Masala raqamini kiriting:'));
if y>=100:
    print("1 dan 99 gacha bo‘lgan sonlarni
kiriting!");
else:
    y=int(y/10);
    m=y%10;
    switch={
        1:"o‘n",
        2:"yigirma",
        3:"o‘ttiz",
        4:"qirq",
        5:"ellik",
        6:"oltmish",
        7:"yetmish",
        8:"sakson",
        9:"to‘qson"
        }
    n=y%10;
    myswitch={
        1:"bir",
        2:"ikki",
        3:"uch",
        4:"to‘rt",
        5:"besh",
        6:"olti",
        7:"yetti",
        8:"sakkiz",
        9:"to‘qqiz"
        }
    print(switch.get(m,''),myswitch.get(n,''),'ta
masala');
```



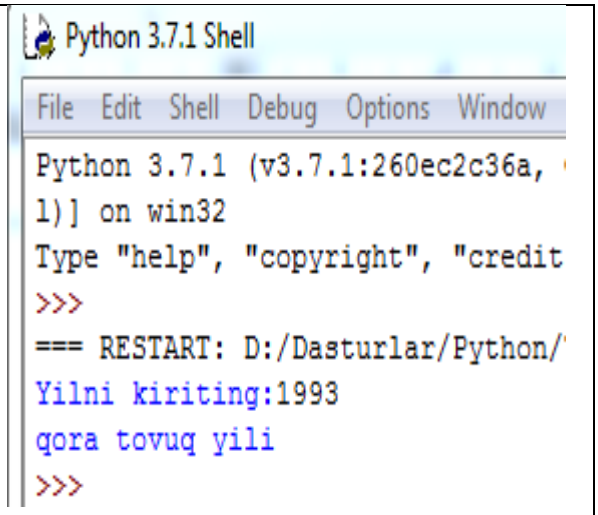
5.18-masala. 100-999 gacha oraliqdagi sonlarni so‘zlarda ifodalovchi dastur tuzing. (masalan: 123-“bir yuz yigirma uch”).

```
y=int(input('Son kiriting:'));
if y>=1000:
    print("1 dan 999 gacha bo‘lgan sonlarni
kiriting!");
else:
    t1=int(y/100);
    my_switch={
        1:"bir yuz",
        2:"ikki yuz",
        3:"uch yuz",
        4:"to'rt yuz",
        5:"besh yuz",
        6:"olti yuz",
        7:"yetti yuz",
        8:"sakkiz yuz",
        9:"to‘qqiz yuz"
        }
    t2=y%100;
    m=int(t2/10);
    switch={
        1:"o‘n",
        2:"yigirma",
        3:"o‘ttiz",
        4:"qirq",
        5:"ellik",
        6:"oltmish",
        7:"yetmish",
        8:"sakson",
        9:"to‘qson"
        }
    t3=int(y/100);
    n=y%10;
    myswitch={
        1:"bir",
        2:"ikki",
        3:"uch",
        4:"to‘rt",
        5:"besh",
        6:"olti",
        7:"yetti",
        8:"sakkiz",
        9:"to‘qqiz"
        }
 print(my_switch.get(t1,''),switch.get(m,''),
myswitch.get(n,''));
```



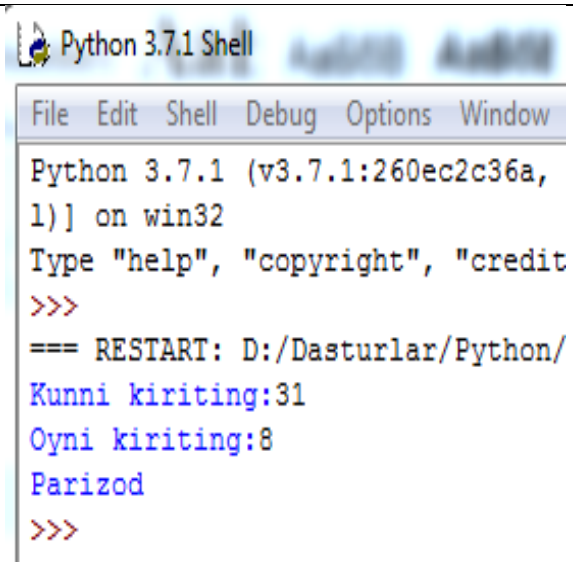
5.19-masala. Sharq kalendarida 60 yillik davr qabul qilingan. Yil muchali 5 ta rang (yashil, qizil, sariq, oq va qora) va 12 ta hayvon (sichqon, sigir, yo‘lbars, quyon, ajdar, ilon, ot, qo‘y, maymun, tovuq, it va to‘ngiz lardan) nomlarining kombinatsiyasidan kelib chiqadi. Yilning raqamiga qarab uning muchalini aniqlovchi dastur tuzing. Masalan: 1984-davr boshi: “Yashil sichqon yili”

```
y=int(input('Yilni kiriting:'));
m=y-3;
n=m%12;
switch={
    0:"to'ng'iz",
    1:"sichqon",
    2:"sigir",
    3:"yo'lbars",
    4:"quyon",
    5:"ajdar",
    6:"ilon",
    7:"ot",
    8:"qo'y",
    9:"maymun",
    10:"tovuq",
    11:"it"
    }
t=m%5;
myswitch={
    0:"qora",
    1:"yashil",
    2:"qizil",
    3:"sariq",
    4:"oq"
    }
print(myswitch.get(t,"),switch.get(n,"), 'yili');
```



5.20-masala. Ikkita burj vaqtlarini aniqlovchi butun son berilgan: D(kun), M(oy). Berilgan sana qaysi burjga kirishini aniqlovchi dastur tuzing."Qovg‘a(20.1-18.2)", "Baliq(19.2-20.3)', "Qoy(21.3-19.4)", "Buzoq(20.4-20.5)", "Egizaklar(21.5-21.6)", "Qisqichbaqa(22.6-22.7)", "Arslon(23.7-22.8)", "Parizod(23.8-22.9)", "Tarozi(23.9-22.10)", "Chayon(23.10-22.11)", "O‘q otar(23.11-21.12)", "Echki(22.12-19.1)".

```
d=int(input('Kunni kiriting:'));
m=int(input('Oyni kiriting:'));
if m==1:
    if d<20:
        print("Echki");
    else:
        print("Qovg'a");
elif m==2:
    if d<19:
        print("Qovg'a");
    else:
        print("Baliq");
elif m==3:
    if d<21:
        print("Baliq");
    else:
        print("Qo'y");
elif m==4:
    if d<20:
        print("Qo'y");
    else:
        print("Buzoq");
elif m==5:
    if d<20:
        print("Buzoq");
    else:
        print("Egizak");
elif m==6:
    if d<22:
        print("Egizak");
    else:
        print("Qisqichbaqa");
elif m==7:
    if d<23:
        print("Qisqichbaqa");
    else:
        print("Arslon");
elif m==8:
    if d<23:
        print("Arslon");
    else:
        print("Parizod");
elif m==9:
    if d<23:
        print("Parizod");
```



```
    else:
        print("Tarozi");
elif m==10:
    if d<23:
        print("Tarozi");
    else:
        print("Chayon");
elif m==11:
    if d<23:
        print("Chayon");
    else:
        print("O'qotar");
elif m==12:
    if d<23:
        print("O'qotar");
    else:
        print("Echki");
else:
    print('Bunday oy mavjud emas!');
```

3.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Topshiriq: 1) Quyidagi topshiriqlarni if…else operatoridan foydalangan holda hisoblash uchun PYTHON tilidagi dasturini tuzing:

4.1-masala. Butun son berilgan. Agar, berilgan son musbat bo‘lsa. 1 ga oshirilsin, aks holda o‘zgartirilmasin. Hosil bo‘lgan sonni ekranga chiqaruvchi dastur tuzing.
4.2-masala. Butun son berilgan. Agar, berilgan son musbat bo‘lsa. 1 ga oshiring, aks holda 2 ga kamaytiring. Hosil bo‘lgan sonni ekranga chiqaruvchi dastur tuzing.
4.3-masala. Butun son berilgan. Agar, berilgan son musbat bo‘lsa. 1 ga oshiring. agar manfiy bo‘lsa 2 ga kamaytiring. Agar 0 ga teng bo‘lsa. 10 ni o‘zlashtirsin. Hosil bo‘lgan sonni ekranga chiqaruvchi dastur tuzing.
4.4-masala. Uchta butun son berilgan. Shu sonlar orasidan nechta musbat son borligini aniqlovchi dastur tuzing.
4.5-masala. Uchta butun son berilgan. Shu sonlar orasidan nechta musbat va manfiy son borligini aniqlovchi dastur tuzing.
4.6-masala. Ikkita butun son berilgan. Shu sonlarning kattasini aniqlovchi dastur tuzing.
4.7-masala. Ikkita butun son berilgan. Shu sonlarning kichigining tartib raqamini aniqlovchi dastur tuzing.
4.8-masala. Ikkita butun son berilgan. Shu sonlarning avval kattasini keyin kichigini ekranga chiqaruvchi dastur tuzing.
4.9-masala. A va B haqiqiy sonlari berilgan. Shu sonlarni shunday o‘zgartirish kerakki, A son kichik B son katta bo‘lsin. A va B ning qiymati ekranga chiqarilsin.

4.10-masala. A va B butun sonlari berilgan. Agar o‘zgaruvchilar o‘zaro teng bo‘lmasa. A va B o‘zgaruvchilari ularning yig'indisini o‘zlashtirsin. Agar teng bo‘lsa. 0 ni o‘zlashtirsin. A va B ning qiymati ekranga chiqarilsin.
4.11-masala. A va B butun sonlari berilgan. Agar o‘zgaruvchilar o‘zaro teng bo‘lmasa. A va B bu sonlarning kattasini o‘zlashtirsin. Agar teng bo‘lsa. 0 ni o‘zlashtirsin. A va B ning qiymati ekranga chiqarilsin.
4.12-masala. Uchta son berilgan. Shu sonlarni kichigini aniqlovchi dastur tuzing.
4.13-masala. Uchta son berilgan. Shu sonlarni o‘ratachasi (ya’ni katta va kichik sonlar orasidagi son) ni aniqlovchi dastur tuzing.
4.14-masala. Uchta son berilgan. Shu sonlarning yig‘indisi eng katta bo‘ladigan ikkitasini ekranga chiqaruvchi dastur tuzing.
4.15-masala. A, B, C haqiqiy sonlari berilgan. Agar berilgan sonlar o‘sish tartibida berilgan bo‘lsa, sonlarni ikkilantiring. aks holda sonlarning ishorasi o‘zgartirilsin. A, B, C ning qiymatlari ekranga chiqarilsin.
4.16-masala. A, B, C haqiqiy sonlari berilgan. Agar berilgan sonlar o‘sish yoki kamayish tartibida berilgan bo‘lsa. sonlarni ikkilantiring. aks holda sonlarning ishorasi o‘zgartirilsin. A. B, C ning qiymatlari ekranga chiqarilsin.
4.17-masala. Uchta butun son berilgan. Shu sonlarni ikkitasi o‘zaro teng. qolgan bittasining tartib raqami aniqlansin.
4.18-masala. To‘rtta butun son berilgan. Shu sonlarni uchtasi o‘zaro teng. qolgan bittasining tartib raqami aniqlansin.
4.19-masala. Sonlar o‘qida uchta A, B, C nuqtalar berilgan. A nuqtaga eng yaqin nuqta va ular orasidagi masofa topilsin.
4.20-masala. Yil berilgan (musbat butun son). Berilgan yilda nechta kun borligini aniqlovchi dastur tuzing. Kabisa yilida 366 kun bor. kabisa bo‘lmagan yilda 365 kun bor. Kabisa yil deb 4 ga karrali yillarga aytiladi. Lekin 100 ga karrali yillar ichida faqat 400 ga karrali bo‘lganlari kabisa yil hisoblanadi. Masalan 300, 1300 va 1900 kabisa yili emas. 1200 va 2000 kabisa yili.

Topshiriq: 2) Quyidagi masalaning tanlash operatoridan foydalanib PYTHON tilidagi dasturini tuzing:

5.1-masala. 0-9 gacha bo‘lgan butun sonlar berilgan. Kiritilgan songa mos ravishda tegishli so‘z bilan ifodalovchi dastur tuzing. (0-nol 1-bir....h.k)
5.2-masala. K butun soni berilgan. Ushbu raqamga mos, rang nomerini chiqaruvchi dastur tuzing.(0-qora, 1-ko‘k, 2- yashil 3- billur, 4- qizil, 5- siyohrang, 6- jigarrang, 7- havorang, 8- sariq, 9-oq). Agar K soni [0,9] oraliqqa tegishli bo'lmasa “xato” so‘zini chop eting.
5.3-masala. Meva nomi berilgan. Kiritilgan meva qaysi turga tegishli ekanligini aniqlovchi dastur tuzing. (Masalan: yong‘oq, “quruq”).
5.4-masala. Davlat nomi berilgan. Shu Davlat qaysi qit’aga tegishli ekanligini aniqlovchi dastur tuzing.
5.5-masala. A, B, C haqiqiy sonlar va amalning bajarilish tartibi raqami berilgan bo‘lsin. A, B va C sonlari ustida arifmetik amallar bajaruvchi dastur tuzing. K-amal quyidagi qiymatlarni qabul qiladi: 1- ko‘paytirish, 2- bo‘lish, 3- qo‘shish, 4- ayirish.

5.6-masala. Uzinlik birliklari quyidagi tartibda berilgan. 1- kilometr, 2-metr, 3-desimetr, 4- santimetr, 5- millimeter. Uzunlik birligini bildiruvchi son (1 - 5 oraliqda) va kesma uzunligi berilgan (haqiqiy son). Kesmaning uzunligini santimetrda ifodalofchi dastur tuzing.

5.7-masala. Og‘irlik birliklari quyidagi tartibda berilgan. 1-tonna, 2-sentner, 3-kilogramm, 4-gramm, 5-milligramm. Og‘irlik birligini bildiruvchi soni va shu birlikdagi og‘irlik qiymati berilgan. Og‘irlikni grammda ifodalovchi dastur tuzing.

5.8-masala. Sanani bildiruvchi to‘rtta butun son berilgan: D1 va D2 (kun) va M1 va M2 (oy), (kabisa bo‘lmagan yil sanasi kiritiladi). Berilgan sanalar oralig‘i necha kun ekanligini ifodalovchi dastur tuzing. Kabisa yilida 366 kun, kabisa bo‘lmagan yilda 365 kun bor bo‘ladi.

5.9-masala. Ikkita butun son berilgan D (kun) va M (oy). (Kabisa bo‘lmagan yil sanasi kiritiladi). Berilgan sanadan oldingi sanani ifodalovchi dastur tuzing.

5.10-masala. Mashina faqat to‘rt ta tomonga ko‘cha oladi(“s”-shimol, “j”-janub, “q”-sharq, “g”-g'arb) va uchta raqamli buyruq: 0-harakatni davom ettir, 1-chapga yur, 2-o‘ngga yur. Y - robot yo‘nalishi va K - buyruq berilgan. Berilgan buyruq bajarilgandan keying mashina holatini aniqlovchi dastur tuzing.

5.11-masala. Samalyot harakatlanayotganda bir tomonga qaratilgan (“s”-shimol, “j”-janub, “q”-sharq, “g”-g'arb) va uchta raqamli buyruq: 0-o‘ngga buril, 1-chapga buril, 2-180° ga burilish. C - samalyotning boshlang‘ich holati va K1, K2 - buyruqlar berilgan. Berilgan buyruq bajarilgandan keyingi samalyot holatini aniqlovchi dastur tuzing.

5.12-masala. Arifmetik progressiyaninmg birinchi va ikkinchi hadi hamda hadlar soni berilgan: 1-ayirmasi d ni 2-n hadini $a_n = a + d \cdot (n-1)$ 3-n ta hadlar yig‘indisi $S_n = \frac{2 \cdot a + d \cdot (n-1) \cdot n}{2}$ topish formulalari. Shu formulalardan bittasi berilganda qolganlarini topuvchi dastur tuzing.

5.13-masala. Kubning qirrasi uzunligi a berilgan: 1-yon sirti $S_{yon}=4*a^2$, 2-to‘la sirti $S_{to‘la}=6*a^2$, 3- hajmi $V=a^3$ ni hisoblash formulalari berilgan. Ulardan bittasini hisoblovchi dastur tuzing.

5.14-masala. Teng tomonli uchburchakning elementlari quyidagi tartibda nomerlangan. Uning 1-tomoni a, 2-ichki chizilgan aylananing radiusi $r = (a * \sqrt{3})/6$, 3 - tashqi chizilgan aylananing radiusi $R = 2 * r$, 4-yuzasi $S = (a^2 * \frac{\sqrt{3}}{4})$ ma’lum. Shu elementlardan bittasi berilganda qolganlarini topuvchi dastur tuzing.

5.15-masala. O‘yin kartasi turlari berilgan 1-gisht, 2-olma. 3-chillak, 4-qarg‘a. 10 lik kartadan katta kartalar quyidagi qiymatlarni o'zlashtirgan: 11-valet, 12-dama, 13-qirol, 14-tuz. Ikkita butun son berilgan N-karta qiymati($6 \le N \le 14$), M-karta turi ($1 \le M \le 14$) kiritilganda karta nomlarini (masalan: “olti qarg‘a”) chiqarib beruvchi dastur tuzing.

5.16-masala. Yoshni yillarda aniqlovchi 1-100 butun sonlar berilgan. Son kiritilganda unga mos so'zlarda ifodalovchi dastur tuzing. (“besh yosh”, “sakson uch yosh” va h.k.)

5.17-masala.. O‘quv masalalarini aniqlovchi 10 000 gacha butun son berilgan. Son kiritilganda unga mos so'zlarda ifodalovchi dastur tuzing. ("yigirmata masala”, “o‘n

uchta masala” va h.k.)

5.18-masala.. [1-9999] gacha oraliqdagi sonlarni so‘zlarda ifodalovchi dastur tuzing. (masalan: 999-“to‘qiz yuz to‘qson to‘qqiz”).

5.19-masala. Sharq kalendarida 60 yillik davr qabul qilingan. Yil muchali 5 ta rang (yashil, qizil, sariq, oq va qora) va 12 ta hayvon (sichqon, sigir, yo‘lbars, quyon, ajdar, ilon, ot, qo‘y, maymun, tovuq, it va to‘ngiz lardan) nomlaring kombinatsiyasidan kelib chiqadi. Yilning raqamiga qarab uning muchalini aniqlovchi dastur tuzing. Masalan: 1984-davr boshi: “Yashil sichqon yili”.

5.20-masala. Joriy sana va ikkita burj vaqtlarini aniqlovchi butun son berilgan: D1 va D2 (kun), M1 va M2 (oy).

Berilgan sanadan ma’lum burjgacha necha kun borligini aniqlovchi dastur tuzing.

"Qovg'a (20.1-18.2)”, “Baliq (19.2-20.3)”, “Qo y (21.3-19.4)”, “Buzoq (20.4-20.5)”, "Egizaklar (21.5-21.6)”, "Qisqichbaqa (22.6-22.7)”, “Arslon (23.7-22.8)”, “Parizod (23.8-22.9)”, “Tarozi (23.9-22.10)”, “Chayon (23.10-22.11)”, “O'qotar (23.11-21.12)”, "Echki (22.12-19.1)”.

IV. BOB. PYTHON DA TAKRORLASH OPERATORLARI

4.1. TAKRORLASH OPERATORI FOR

Bir xil hisoblash bloklarining bir necha bor takrorlanuvchi jarayoniga takrorlanish (sikl) deyiladi. PYTHON tilida takrorlanish operatorining ikki xil turi mavjud:

1. **for** takrorlanish operatori;
2. **while** takrorlanish operatori;

Yechilayotgan masalaga qarab, dasturchi o‘zi uchun qulay bo‘lgan takrorlanish operatoridan foydalanishi mumkin.
for takrorlash operatorining sintaksisi quyidagicha:

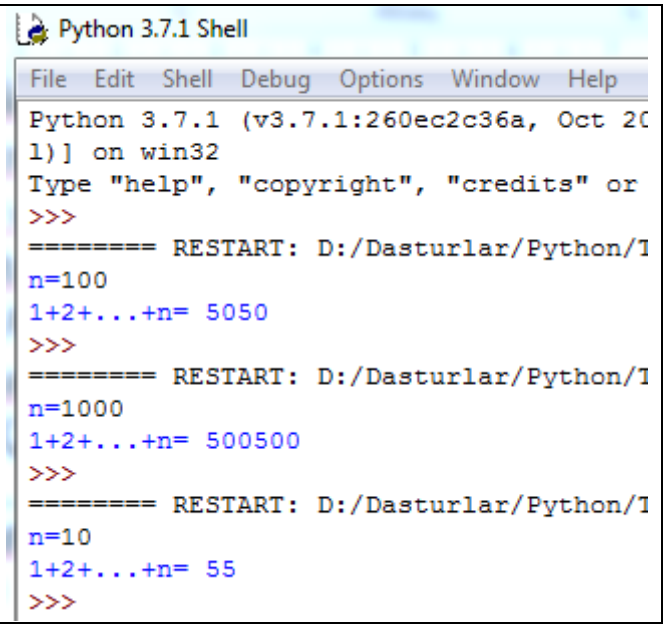
for i in range(a, b, d):
<operator yoki blok>;

Bu yerda **i** takrorlanishlar soni, **a** takrorlanishning birinchi qiymati, **b** takrorlanishning oxirgi qiymati, **d** qadam qiymati., takrorlanish tanasi - <operator yoki blok> bajariladi va oxirida <ifoda3> bajariladi, aks holda boshqaruv takrorlash operatoridan keyingi operatorga o‘tiladi. Takrorlanish tanasi – <operator yoki blok> sifatida bitta operator, shu jumladan bo’sh operator, yoki operatorlar bloki kelishi mumkin.
Takrorlanish takrorlanishi davomida bajarilishi lozim bo'lgan operatorlar majmuasi takrorlanish tanasi deyiladi. Takrorlanish tanasi sifatida bir yoki bir nechta operatordan foydalanish mumkin.

Agar takrorlanish tanasida bir nechta operatordan foydalanmoqchi bo`lsak bu operatorlarni blok { } orasiga olishimiz kerak.

Quyidagi masalada 1 dan n gacha sonlarning yig‘indisini hisoblaymiz:

```
n=int(input('n='))
S=0
for i in range(1,n+1):
    S+=i
print("1+2+...+n=",S)
```

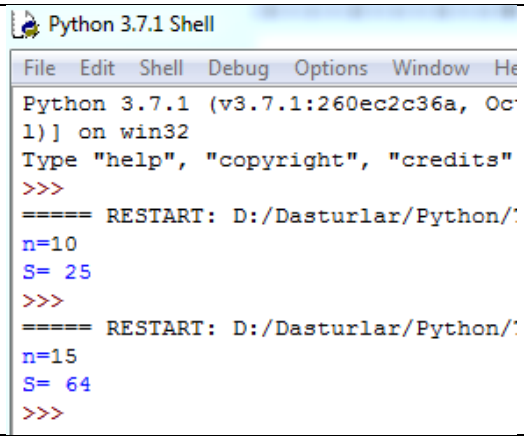


Natija oldingi rasmda ko‘rsatilganiga o‘xshash.

4.2. PYTHON DA FOR TAKRORLASH OPERATORI TADBIQI

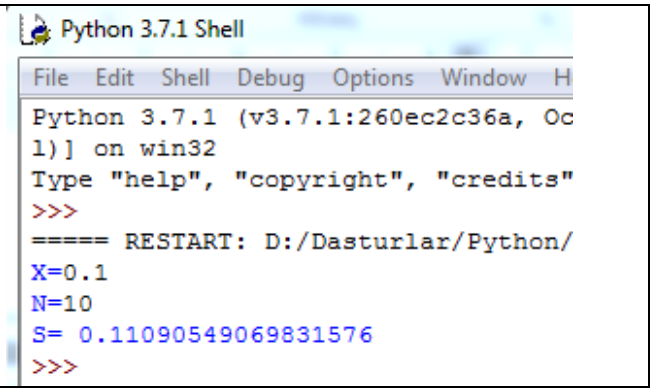
6.1-masala. 1 dan n gacha bo‘lgan sonlarning faqat toq raqamlarining yig‘indisini hisoblovchi dastur tuzing.

```
n=int(input('n='))
S=0
for i in range(1,n+1):
    if i%2==1:
        S+=i
print('S=',S)
```



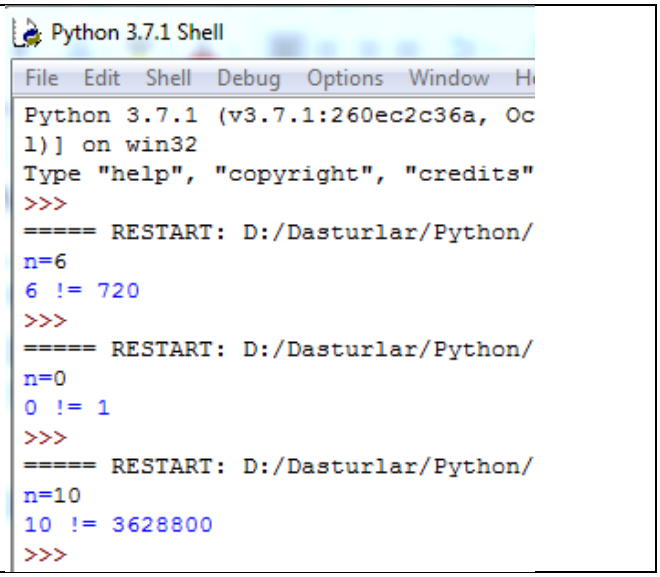
6.2-masala. N natural son va X haqiqiy sonlar berilgan. Quyidagi yig‘indini hisoblang. $S = \sin X + \sin^2 X + ... + \sin^N X$. Yechish. Izlanayotgan yig’indini S bilan belgilaymiz.

```
import math
x=float(input('X='))
n=int(input('N='))
s=0
x=math.sin(x)
for i in range(1,n+1):
    s=s+math.pow(x,i)
print('S=',s)
```



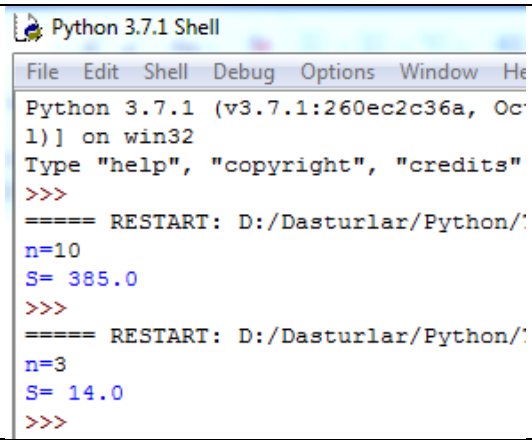
6.3-masala. N! hisoblash talab qilingan bo‘lsin, bunda N natural son. Yechish. N<34 bo‘lganda natural sonlar faktorialini hisoblash mumkin.

```
n=int(input('n='));
p=1;
for i in range(1,n+1):
    p=p*i;
print(n,'!=',p)
```



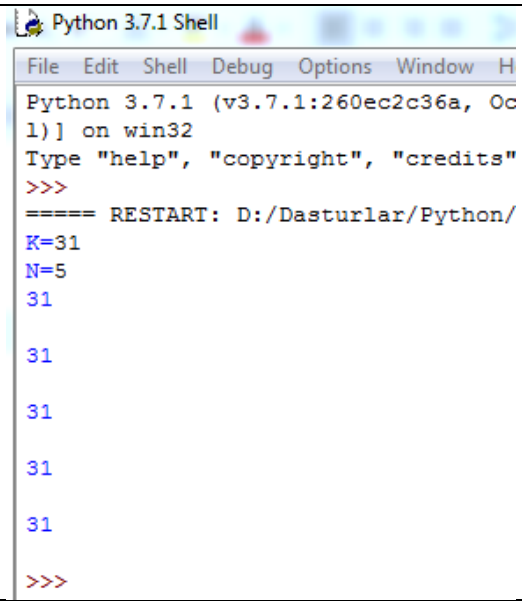
6.4-masala. 1 dan n gacha bo‘lgan natural sonlar kvadratlari yig‘indisini toping. Yechish. Izlanayotgan yig‘indini S bilan belgilaymiz.

```
import math
n=int(input('n='));
s=0;
for i in range(1,n+1):
    s=s+math.pow(i,2);
print('S=',s)
```



6.5-masala. k va n butun sonlari berilgan (n>0). κ sonini n marta chiqaruvchi dastur tuzing.

```
k=int(input('K='));
n=int(input('N='));
for i in range(1,n+1):
    print(k,'\n');
```

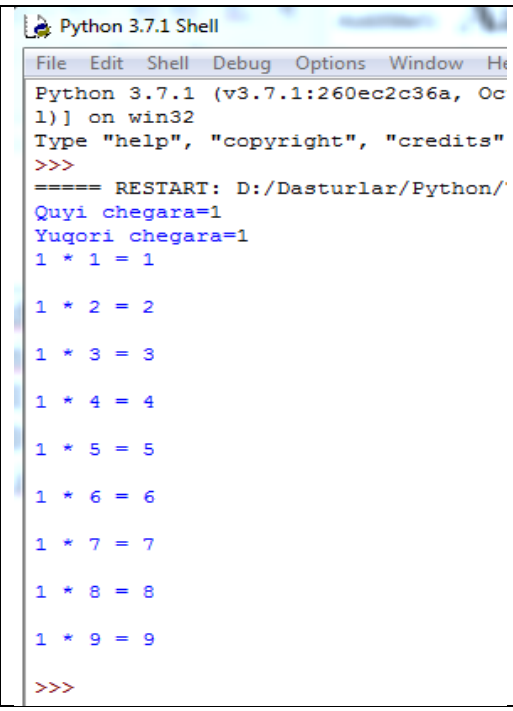


6.6-masala. 1 dan n gacha bo‘lgan sonlardan sikl qadami 1 ga teng holda kvadrat ildiz chiqaring. Yechish. Berilgan x sondan chiqarilgan kvadrat ildizning qiymatini y bilan belgilaymiz: $y=\sqrt{x}$.

```
import math
n=int(input('n='));
y=0;
for i in range(1,n+1):
    y=y+math.sqrt(i);
print('Y=',y);
```

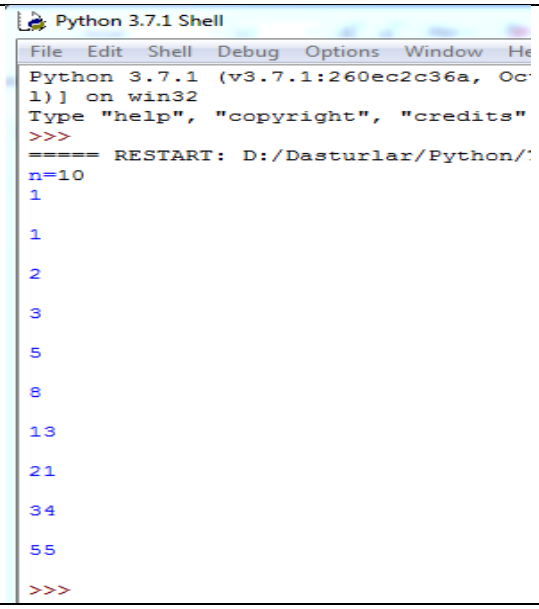
6.7-masala. 1 dan 9 gacha bo‘lgan sonlarni ko’paytirish jadvalini ekranga chiqaring. Yechish. Bu masalani yechish uchun 3 marta sikl buyrug‘idan foydalanamiz. Birinchi siklda birinchi ko‘paytuvchi 1 dan 3 gacha, ikkinchisi esa, 1 dan 9 gacha o‘zgaradi. Ikkinchisi siklda birinchi ko‘paytuvchi 4 dan 6 gacha, ikkinchisi esa, 1 dan 9 gacha o’zgaradi. Uchinchi siklda birinchi ko‘paytuvchi 7 dan 9 gacha, ikkinchisi esa, 1 dan 9 gacha o‘zgaradi.

```
a=int(input('Quyi chegara='));
b=int(input('Yuqori chegara='));
for i in range(a,b+1):
    for j in range(1,10):
        print(i,'*',j,'=',i*j,'\n');
```



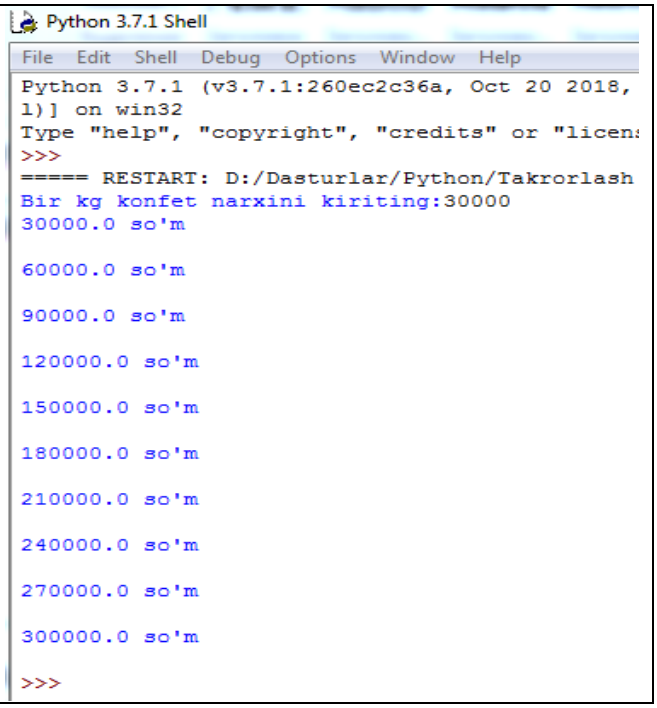
6.8-masala. L nomerli Fibonachchi sonini ekranga chiqaring. Yechish. 1,1,2,3,5,8,13,21,34,...sonlar Fibonachchi sonlar ketma-ketligini ifodalaydi. Bu sonlar ketma-ketligida uchinchi hadidan boshlab har bir son o‘zidan oldingi ikkita sonning yig‘indisiga teng.

```
n=int(input('n='));
a1=1;
a2=1;
for i in range(1,n+1):
    if i==1 or i==2:
        a=1;
    else:
        a=a1+a2;
        a1=a2;
        a2=a;
    print(a,'\n');
```



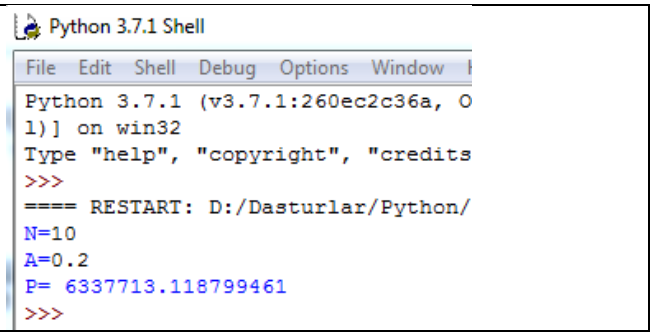
6.9-masala. Bir kilogramm konfetning narxi berilgan (haqiqiy son), 1, 2, ...,10 kg konfetning narxini chiqaruvchi dastur tuzing.

```
n=float(input("Bir kg konfet narxini
kiriting:"));
for i in range(1,11):
    s=i*n;
    print(s,"so'm\n");
```



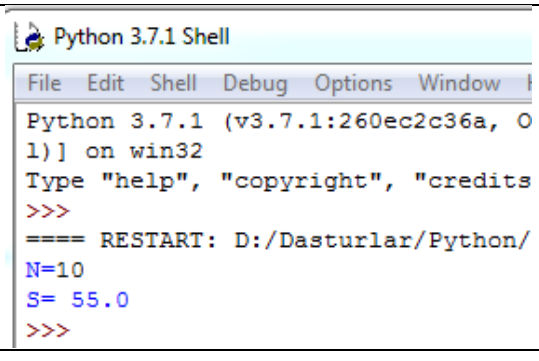
6.10-masala. N natural son va A haqiqiy son berilgan. Quyidagi ko‘paytmani hisoblang: p=A(A+1)(A+2)...(A+N). Yechish. Berilgan ko‘paytmani p bilan belgilaymiz.

```
n=int(input('N='))
a=float(input('A='));
p=1;
for i in range(1,n+1):
    p=p*(a+i);
print('P=',p);
```



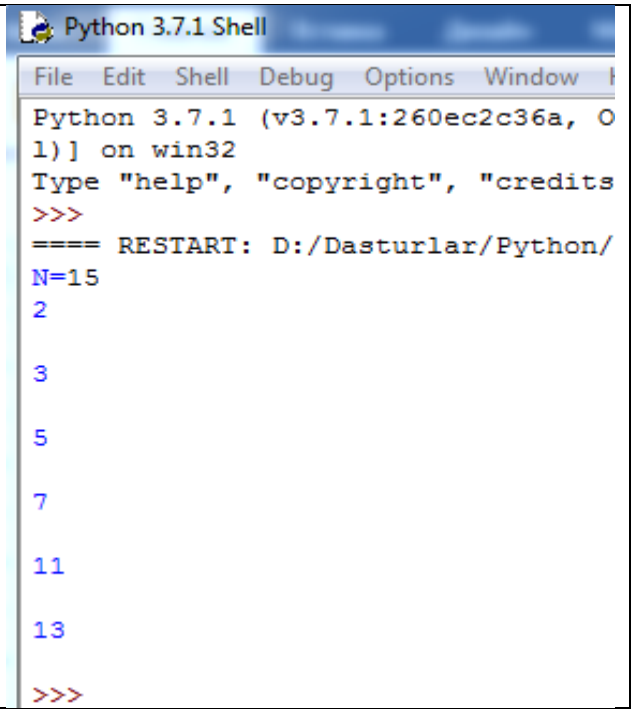
6.11-masala. Darajaga ko‘tarish amalini bajarmay, ushbu yig‘indini hisoblang: $S=\sum_{n=1}^{10}(-1)^n n^2$. Yechish. Bu masalani yechishda $(-1)^n$ ni hisoblash uchun yangi o‘zgaruvchi c=1 ni kiritamiz. Uning har galgi qiymatini -1 ga ko‘paytiramiz.

```
import math;
n=int(input('N='));
c=-1;
s=0;
for i in range(1,n+1):
    s=s+c*math.pow(i,2);
    c=-1*c;
print('S=',s);
```



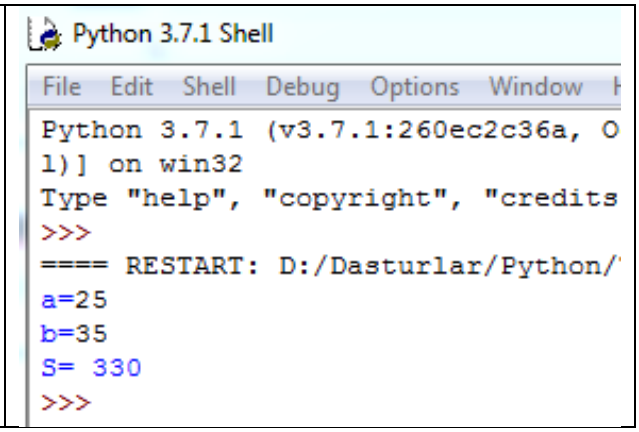
6.12-masala. Natural sonni tub ko‘paytuvchilarga ajratish dasturi.

```
n=int(input('N='));
for i in range(2,n+1):
    tub=True;
    for j in range(2,int(i/2)+1):
        if i%j==0:
            tub=False;
            break;
    if tub:
        print(i,'\n');
```



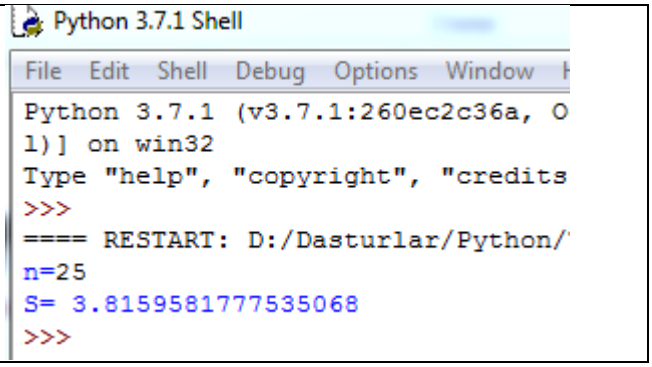
6.13-masala. a va b butun sonlari berilgan (a<b), a dan b gacha bo‘lgan barcha butun sonlar yig‘indisini chiqaruvchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
s=0;
if a<b:
    for i in range(a,b+1):
        s=s+i;
print('S=',s);
```



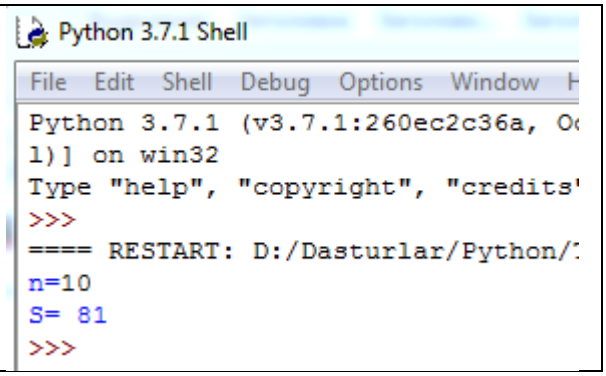
6.14-masala. n butun soni berilgan (n>0). Quyidagi yig'indini hisoblovchi dastur tuzing: S=1+1/2+1/3+...+1/n.

```
n=int(input('n='));
s=0;
for i in range(1,n+1):
    s=s+1/i;
print('S=',s);
```



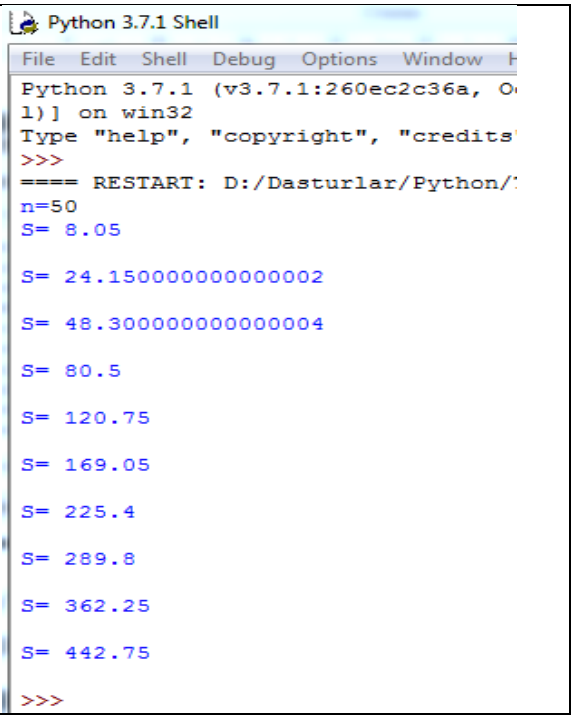
6.15-masala. n butun soni berilgan (n>0). Shu sonning kvadratini quyidagi formula asosida hisoblovchi dasturini tuzing. S=1+3+5+...+(2*n-1).

```
n=int(input('n='));
s=0;
for i in range(1,n):
    s=s+(2*i-1);
print('S=',s);
```



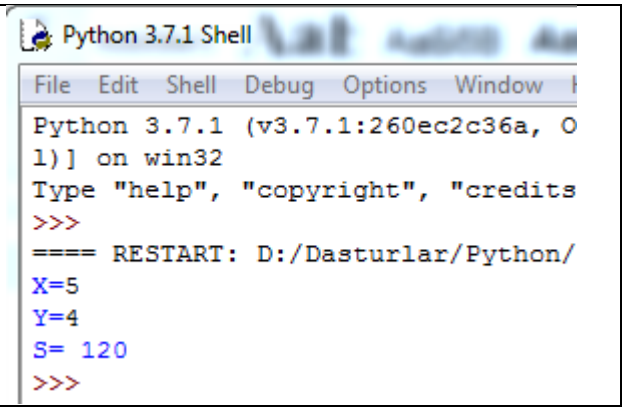
6.16-masala. 5 mildan 50 milgacha masofani 5 ga teng qadam bilan kilometrga o‘tkazing.

```
n=int(input('n='));
s=0;
for i in range(5,n+1,5):
    s=s+i*1.61;
    print('S=',s,'\n');
```



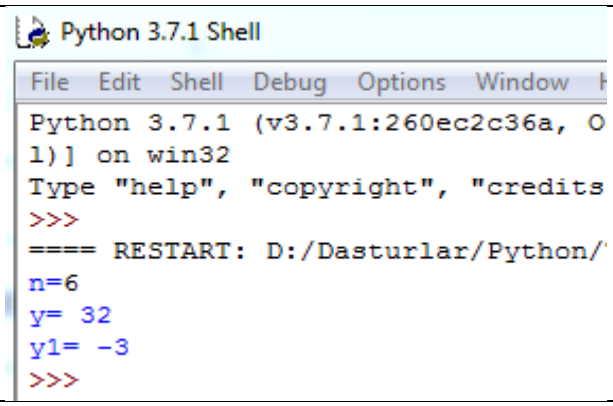
6.17-masala. Tomonlari X va Y ga teng to‘rtburchak yuzini X=3 dan 5 gacha Y=1 dan 4 gacha 1 ga teng qadam bilan o‘zgarganda hisoblang.

```
x=int(input('X='));
y=int(input('Y='));
s=0;
for i in range(3,x+1):
    for j in range(1,y+1):
        s=s+i*j;
print('S=',s);
```



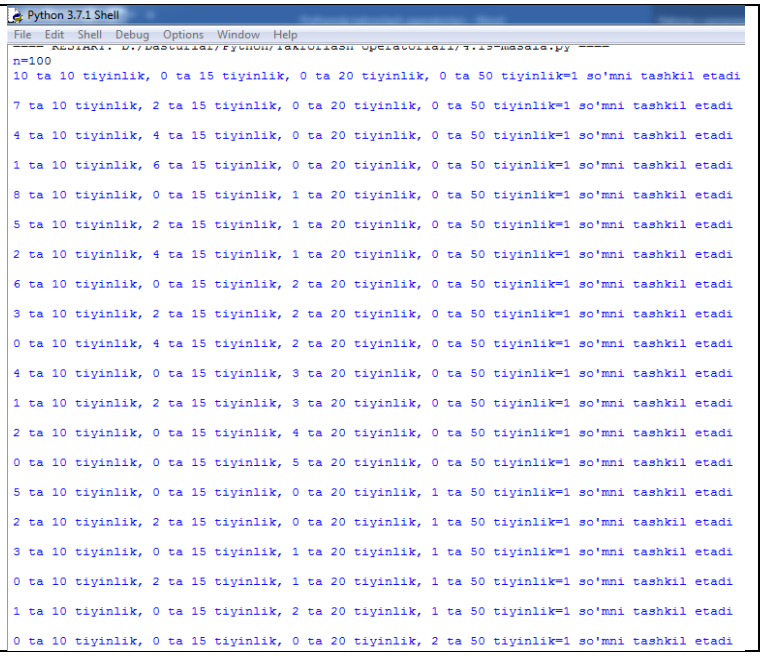
6.18- masala. $y = 3x^2 + 4x - 10$ funksiyaning x o‘zgaruvchi 1) 0,1,2,3,4,5; 2) 0,3,6,9,12 ga teng qiymatlarni qabul qilgandagi ifodalarni hisoblash dasturi.

```
n=int(input('n='));
for i in range(n+1):
    if i-1:
        y=3*i+4*i-10;
    elif 3*i:
        y1=3*i+4*i-10;
print('y=',y);
print('y1=',y1);
```



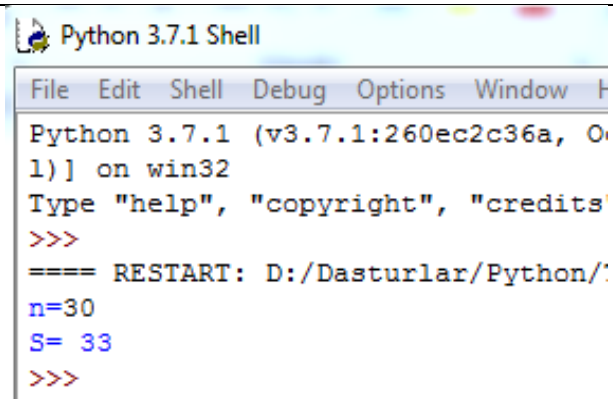
6.19-masala. 1 so‘mni 50, 20, 15, 10 tiyinlik tangalar bilan qanday usullar bilan maydalash mumkin?

```
n=int(input('n='));
for l in range(3):
    for k in range(6):
        for j in range(8):
            for i in range(11):

s=i*10+j*15+k*20+l*50;
                if s==n:
                    print(i,'ta 10
tiyinlik,',j,'ta 15 tiyinlik,',k,'ta 20
tiyinlik,',l,"ta 50 tiyinlik=1 so'mni
tashkil etadi\n");
```



6.20-masala. N butun soni berilgan. Quyidagi yig‘indini chiqaruvchi dastur tuzing. S=11+22+...+NN.

```
import math;
n=int(input('n='));
s=0;
for i in range(1,n+1):
    m=i%10;
    y=math.floor(i/10);
    m1=y%10;
    if m==m1:
        s=s+i;
print('S=',s);
```



4.3. TAKRORLANUVCHI *WHILE* OPERATOR

Operator ***while*** shartli sikl operatori deyiladi, siklga kirishda oldin shartli ifoda hisoblanadi, agar uning qiymati noldan farqli bo‘lsa sikl tanasi bajariladi. Shundan so‘ng shartli ifodani hisoblash va sikl tanasi operatorlarini bajarish, shartli ifoda qiymati nolga teng bo‘lguncha davom etadi. Takrorlanishlar soni oldindan aniq bo'lmaganda va qandaydir shartga bog'liq bo'lganda ***while*** operatoridan foydalanamiz. ***While*** takrorlash operatorining sintaksisi quyidagicha:

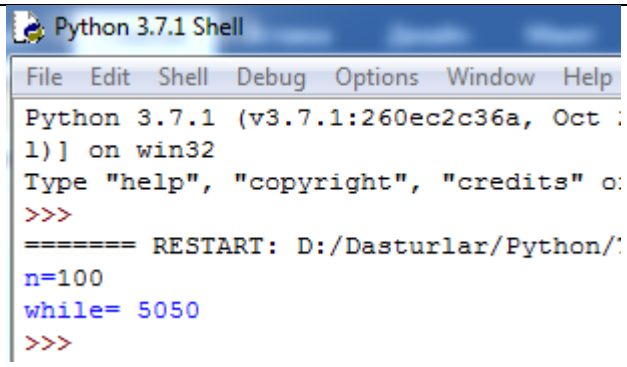
while <shart>:

<operatorlar>

Bu yerda shart rost bo‘lganda operatorlar qismi bajariladi.

Quyidagi masalada 1 dan n gacha sonlarning yig‘indisini while da hisoblaymiz:

```
n=int(input('n='));
s=0;
i=1;
while i<=n:
    s=s+i;
    i=i+1;
print('while=',s);
```



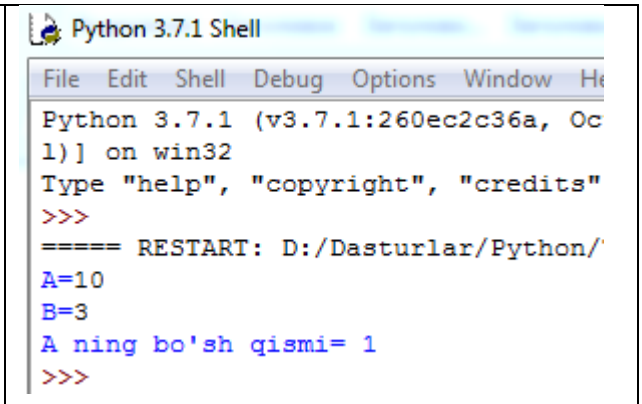
Yuqoridagi dasturni tahlil qilamiz:

- **Avval int toifasidagi**

4.4. TAKRORLANUVCHI *WHILE* OPERATORI TADBIQI

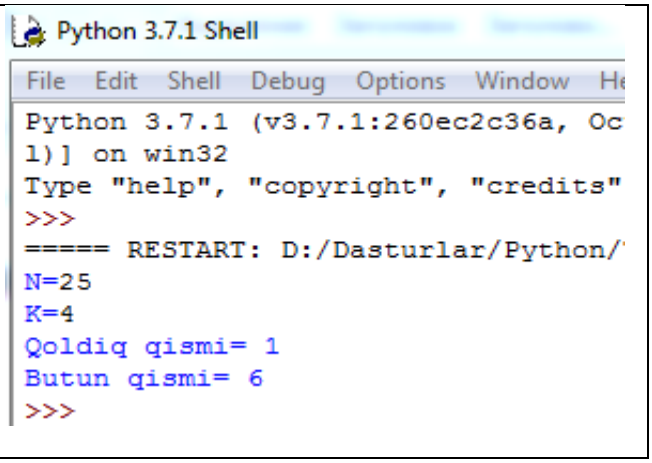
7.1-masala. A va B butun musbat sonlari berilgan (A>B). A usunlikdagi kesmada maksimal darajada B kesma joylashtirilgan. A kesmaning bo‘sh qismini aniqlovchi dastur tuzing. Ko‘paytirish va bo‘lish amallarini ishlatmang.

```
a=int(input('A='));
b=int(input('B='));
while b<a:
    a=a-b;
print("A ning bo'sh qismi=",a);
```



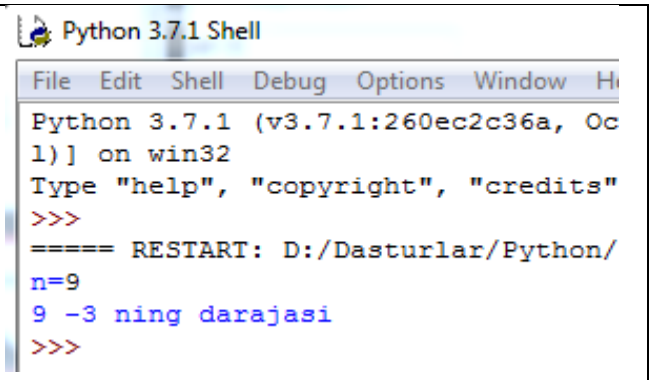
7.2-masala. N va K butun musbat sonlari berilgan. Faqat ayirish va qo‘shish amallarini ishlatib N sonini K soniga bo‘lgandagi qoldiq va butun qismini aniqlovchi dastur tuzing.

```
N=int(input('N='));
K=int(input('K='));
butun=0;
while K<N:
    N=N-K;
    butun+=1;
print('Qoldiq qismi=',N);
print('Butun qismi=',butun);
```



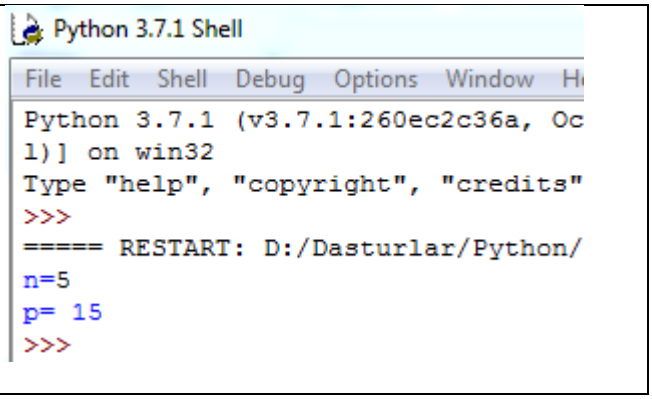
7.3-masala. n butun soni berilgan (n > **0).** Agar n soni **3** ning darajasi bo‘lsa ***3** - ning darajasi’. aks xolda ***3** - ning darajasi emas” degan natija chiqaruvchi dastur tuzing. Qoldiqli bo‘lish va bo‘lish amallarini ishlatmang.

```
n=int(input('n='));
i=1;
while i<n:
    i*=3;
if n==i:
    print(n,'-3 ning darajasi');
else:
    print(n,'-3 ning darajasi emas');
```



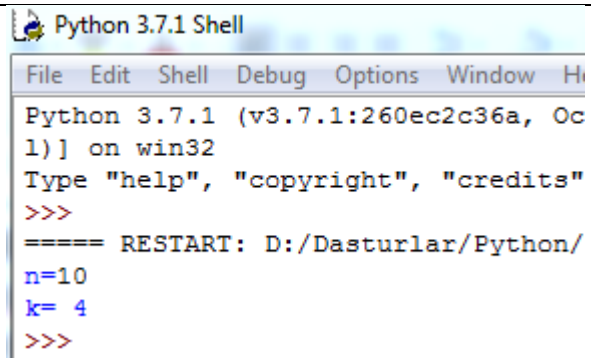
7.4-masala. n natural soni berilgan (n > 0**).** Quyidagi ifodani hisoblovchi dastur tuzing: n!! = n * (n - 2) * (n - 4**).** Agar n juft bo‘lsa oxirgi ko‘paytuvchi 2, toq bo‘lsa 1 bo'ladi.

```
n=int(input('n='));
p=1;
while 2<=n:
    p=p*n;
    n=n-2;
print('p=',p);
```



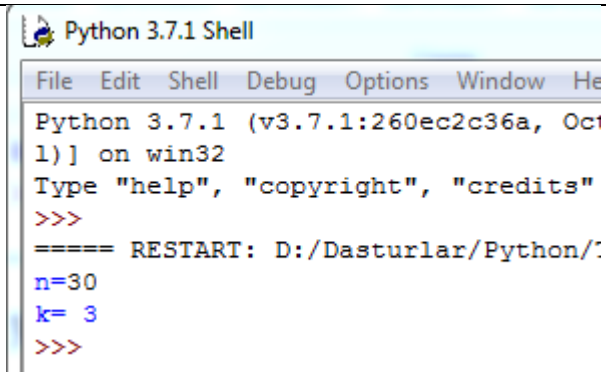
7.5-masala. n natural soni berilgan (n>0**).** Kvadrati n dan katta bo‘ladigan eng kichik butun κ sonini (k^2>n) aniqlovchi dastur tuzing. Ildizdan chiqaruvchi funksiyadan foydalanmang.

```
import math;
n=int(input('n='));
k=0;
while (math.pow(k,2)>n)==False:
    k+=1;
print('k=',k);
```



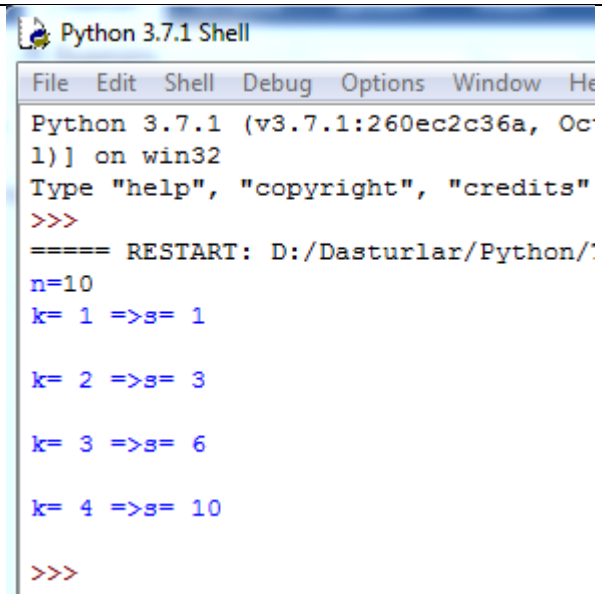
7.6-masala. n natural soni berilgan (n>1**).** 3^k<=n shartni qanoatlantiruvchi eng katta butun κ sonini aniqlovchi dastur tuzing.

```
n=int(input('n='));
k=0;
while n>=3:
    n/=3;
    k+=1;
print('k=',k);
```



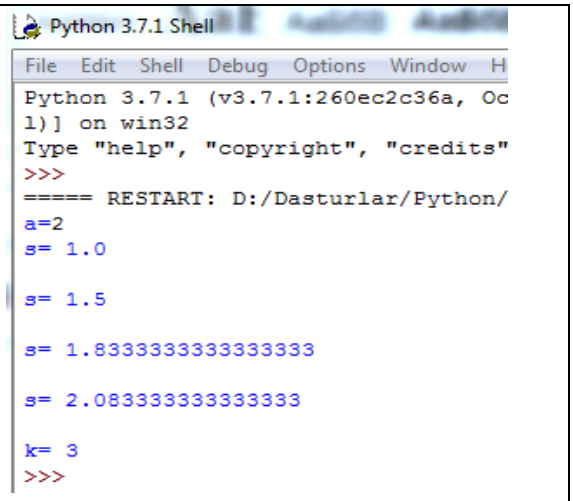
7.7-masala. n natural soni berilgan (n>1). (1+2+3+...+k)>=n shart bajariladigan eng kichik κ sonini aniqlovchi dastur tuzing. 1 dan κ gacha bo‘lgan yig‘indi ham ekranga chiqarilsin.

```
n=int(input('n='));
k=0;
s=0;
while n>s:
    k=k+1;
    s=s+k;
    print('k=',k,'=>s=',s,'\n');
```



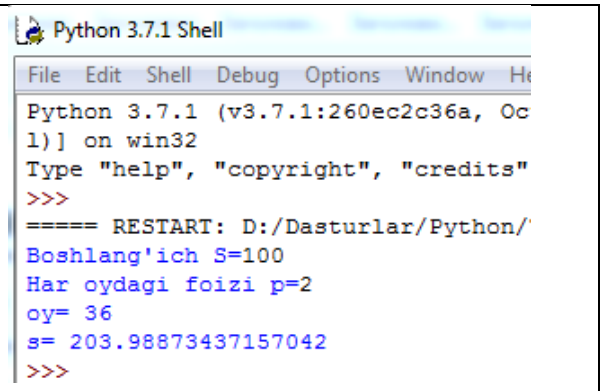
7.8-masala. a soni berilgan (a>1). (1+1/2+1/3+...+1/k)<=a shart bajariladigan eng katta к sonini aniqlovchi dastur tuzing. Yig‘indi ham ekranga chiqarilsin.

```
a=int(input('a='));
k=0;
s=0;
while a>s:
    k=k+1;
    s=s+1/k;
    print('s=',s,'\n');
if s>a:
    s-=1/k;
    k-=1;
print('k=',k);
```



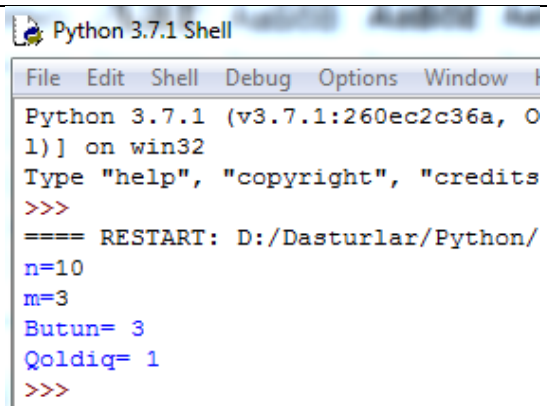
7.9-masala. Bankka boshlang‘ich S so‘m qo‘yildi. Har oyda bor bo‘lgan summa p foizga oshadi (0<p<25). Necha oydan keyin boshlang‘ich qiymat 2 martadan ko‘p bo‘lishini hisoblovchi dastur tuzing. Necha oy к-butun son. Bankda hosil bo‘ladigan summa haqiqiy son ekranga chiqarilsin.

```
s=int(input("Boshlang'ich S="));
p=int(input("Har oydagi foizi p="));
oy=0;
b=2*s;
while b>s:
    s+=s*p/100;
    oy+=1;
print('oy=',oy,'\ns=',s);
```



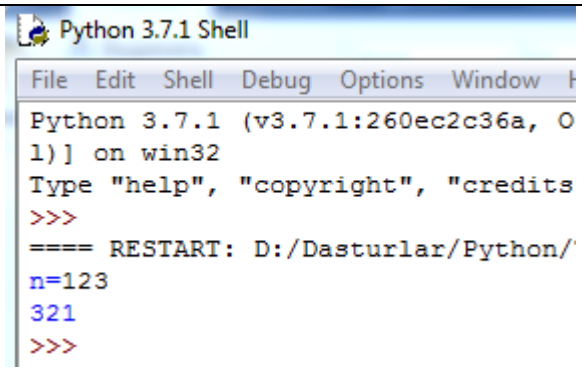
7.10-masala. n va m butun musbat sonlari berilgan (n>m). n sonini m soniga bo‘lib butun hamda qoldiq qismlarini bo‘lish va qoldiqni olish amallarini ishlatmasdan topuvchi dastur tuzing.

```
n=int(input('n='));
m=int(input('m='));
butun=0;
while n>m:
    n=n-m;
    butun+=1;
print("Butun=",butun,"\nQoldiq=",n);
```



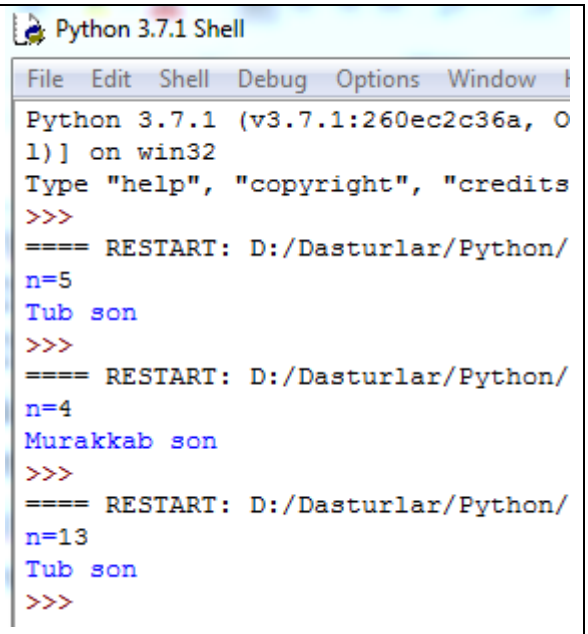
7.11-masala. n butun soni berilgan (n>0). Uni bo‘lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarini teskari tartibda chiqaruvchi dastur tuzing.

```
import math;
n=int(input('n='));
while n>0:
    i=n%10;
    n=math.floor(n/10);
    print(i, end='');
```



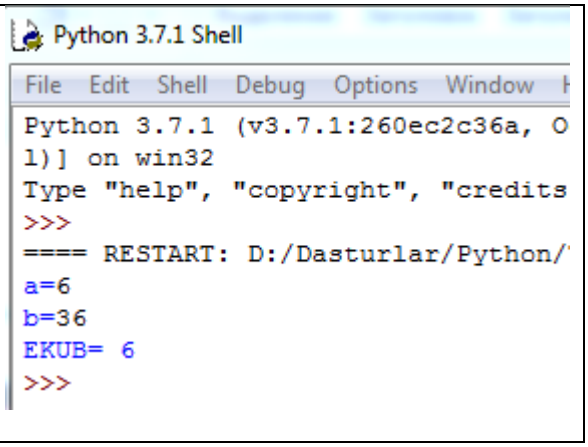
7.12-masala. n butun soni berilgan (n>1). n sonining tub yoki tub emasligini aniqlovchi dastur tuzing.

```
n=int(input('n='));
i=2;
j=2;
while i<=n:
    tub=True;
    i=i+1;
    while j<=i/2:
        if i%j==0:
            tub=False;break;
        j=j+1;
if tub==False:
    print("Tub son");
else:
    print("Murakkab son");
```



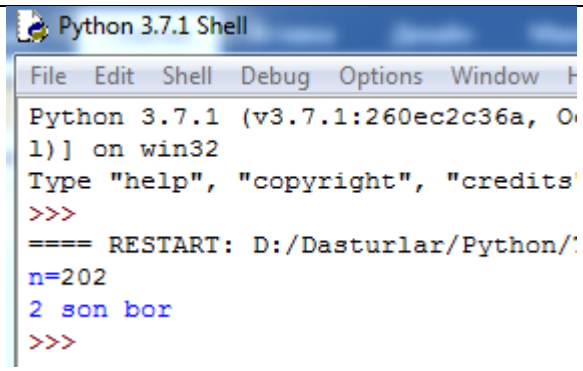
7.13-masala. a va b butun musbat sonlari berilgan. Berilgan sonlarning eng katta umumiy bo‘luvchisini aniqlovchi dastur tuzing.

```
a=int(input('a='));
b=int(input('b='));
while a!=b:
    if a>b:
        a=a-b;
    else:
        b=b-a;
print('EKUB=',a);
```



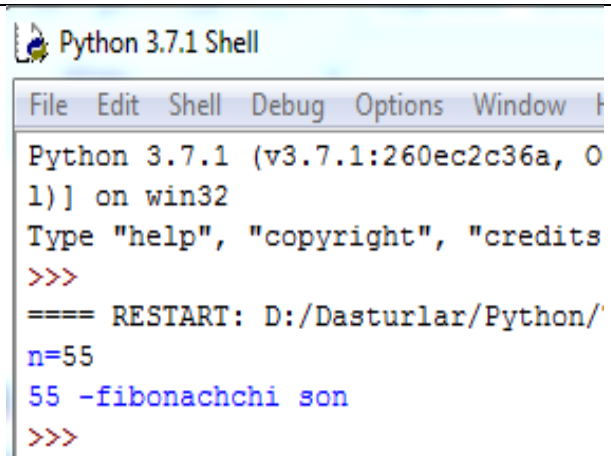
7.14-masala. n butun soni berilgan (n>0**)**. Uni bo‘lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlarining orasida 2 raqami bor yoki yo‘qligini aniqlovchi dastur tuzing.

```
n=int(input('n='));
bor=False;
while n>0:
   x=n%10;
   n=n/10;
   if x==2:
      bor=True;
if bor==True:
   print("2 son bor");
else:
   print("2 son yo'q");
```



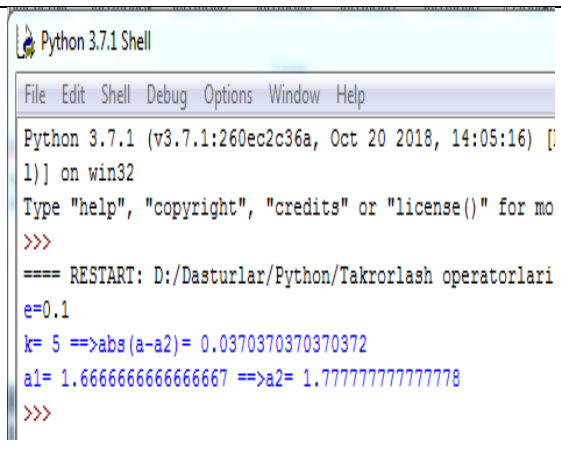
7.15-masala. n butun soni berilgan (n>1**).** n sonini Fibonachchi sonlari orasida bor yoki yo‘qligini aniqlovchi dastur tuzing. Fibonachchi sonlari quyidagi qonuniyat asosida topiladi. $F_1=1$; $F_2=1$; $F_k=F_{k-2}+F_{k-1}$; $\kappa=3,4,...$

```
n=int(input('n='));
f1=1;
f2=1;
f3=0;
while f3<n:
   f3=f1+f2;
   f1=f2;
   f2=f3;
if n==f3:
   print(n,'-fibonachchi son');
else:
   print(n,'-fibonachchi son emas');
```



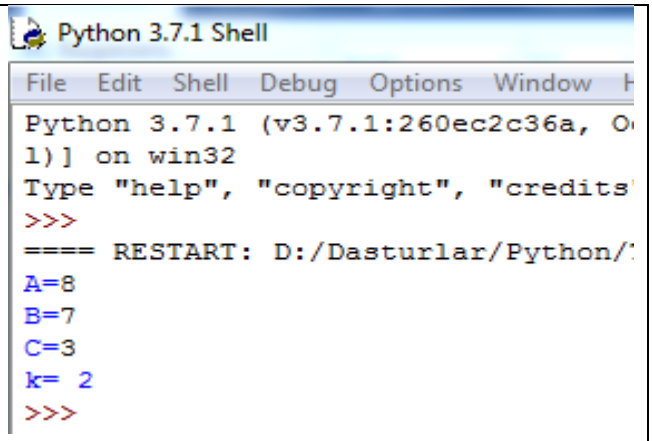
7.16-masala. e haqiqiy musbat soni berilgan. Ketma - ketlik xadlari quyidagicha aniqlanadi: $a_1=1$; $a_2=2$; $a_k=(a_{k-2}+2*a_{k-1})/3$; $\kappa=3,4,...$; $|a_k-a_{k-1}|<e$ shartni qanoatlantiruvchi eng kichik κ sonini aniqlovchi dastur tuzing. a_k va a_{k-1} ham ekranga chiqarilsin.

```
e=float(input('e='));
a1=1;
a2=2;
k=2;
while 1:
   a=(a1+2*a2)/3;
   k+=1;
   if abs(a-a2)<e:
      y=abs(a-a2);
      print('k=',k,'==>abs(a-a2)=',y);
      print('a1=',a1,'==>a2=',a2);break;
   a1=a2;   a2=a;
```



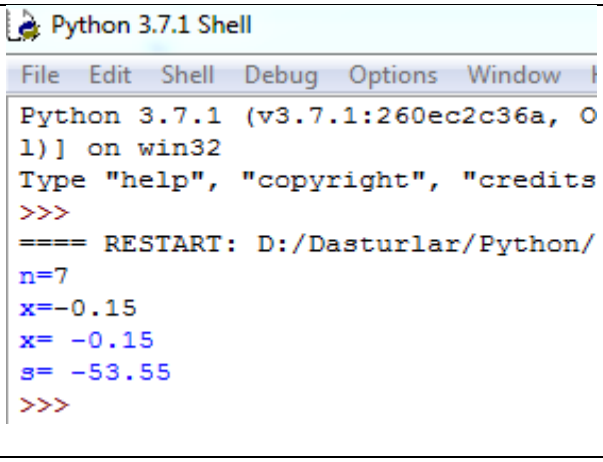
7.17-masala. A, B, C musbat butun sonlari berilgan. AxB to‘rtburchak ichida tomoni C bo‘lgan kvadratdan nechtasi joylashishini aniqlovchi dastur tuzing. Ko‘paytirish va bo‘lish amallarini ishlatmang.

```
A=int(input('A='));
B=int(input('B='));
C=int(input('C='));
k=0;
while A>=C and B>=C:
    A-=C;
    B-=C;
    k+=1;
print('k=',k);
```



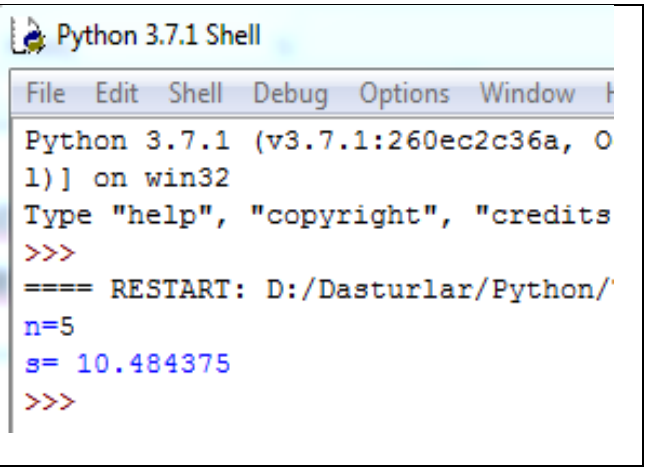
7.18-masala. n butun soni va x haqiqiy soni berilgan (n>0, |x|<1). Quyidagi yig‘indini hisoblovchi dastur tuzing:
x+1*x*3+1*3*x*5+...+1*3*...*(2*n-1)*x*(2n+1)

```
n=int(input('n='));
x=float(input('x='));
s=0;
i=0;
while i<n:
    if abs(x)<1:
        s=s+((2*i-1)*x*(2*i+1));
    i=i+1;
print('x=',x,'\ns=',s);
```



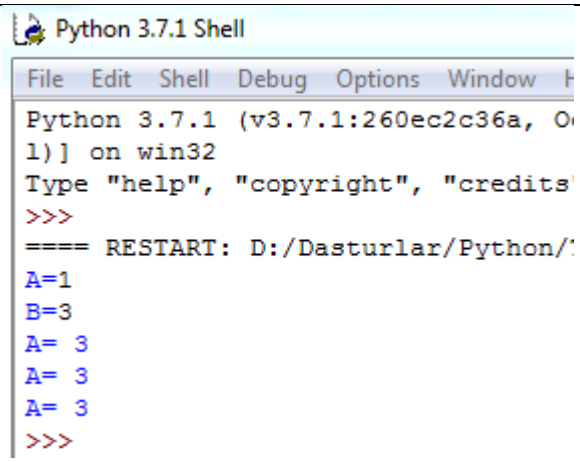
7.19-masala. n butun soni berilgan (n>0). Quyidagi ketma - ketlikning dastlabki n ta hadini chiqaruvchi dastur tuzing: A(0)=2; $A(K)=2+1/A(K+1)$; $K=1,2,$ ...

```
n=int(input('n='));
s=0;
k=0;
A=2;
while k<n:
    A+=A;
    s=s+2+1/A;
    k=k+1;
print('s=',s);
```



7.20-masala. A va B butun soni berilgan (A<B). A va B sonlari orasidagi barcha butun sonlarni chiqaruvchi dastur tuzing. Bunda har bir son o‘zining qiymaticha chiqarilsin, ya‘ni 3 soni 3 marta chiqariladi.

```
A=int(input('A='));
B=int(input('B='));
n=0;
while B>n:
    if A<B:
        A=B;
    print('A=',A);
    n+=1;
```



4.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Topshiriq: 1) Quyidagi masalalarning PYTHON tilidagi dasturini for takrorlash operatoridan foydalanib tuzing:

6.1-masala. n butun soni berilgan (n > 0). Bir sikldan foydalangan holda quyidagi yig‘indini hisoblovchi dastur tuzing. (Olingan natija taxminan e=exp(1) ga yaqinlashadi). 1 + 1/(1!) + 1/(2!) + 1/(3!) + ... +1/(n!)

6.2-masala. n butun soni va x haqiqiy soni berilgan (n>0). Quyidagi yig‘indini hisoblovchi dastur tuzing. (Olingan natija taxminan e^x ga yaqinlashadi). $1+x+x^2/(2!)+x^3/(3!)+...+x^n/(n!)$.

6.3-masala. n butun soni va x haqiqiy soni berilgan (n>0). Quyidagi yig‘indini hisoblovchi dastur tuzing. (Olingan natija taxminan sin(x) ga yaqinlashadi) $x-x^3/(3!)+x^5/(5!)-...+(-1)^n\ x^{2n+1}/((2*n+1)!)$.

6.4-masala. n butun soni va x haqiqiy soni berilgan (n>0). Quyidagi yig‘indini hisoblovchi dastur tuzing. (Olingan natija taxminan cos(x) ga yaqinlashadi) $1-x^2/(2!)+x^4/(4!)-...+(-1)^n\ x^{2n}/((2*n)!)$.

6.5-masala. n butun soni va x haqiqiy soni berilgan (n>0, |x|<1). Quyidagi yig‘indini hisoblovchi dastur tuzing. $x-x^2/2+x^3/3-...+(-1)^n\ x^n/n$.

6.6-masala. n butun soni va x haqiqiy soni berilgan (n>0. |x|<1). Quyidagi yig‘indini hisoblovchi dastur tuzing. $x-x^3/3+x^5/5-...+(-1)^n\ x^{2n+1}/(2n+1)$.

6.7-masala. n butun soni va x haqiqiy soni berilgan (n>0. |x|<1). Quyidagi yig‘indini hisoblovchi dastur tuzing. $1+x/2-1*x^2/(2*4)+1*3*x^3/(2*4*6)-...+(-1)^n1*1*3*...*(2*n-3)*x^n/(2*4*..*(2*n))$.

6.8-masala. n butun soni va sonlar o‘qida 2 ta A. B nuqta berilgan. (A, B haqiqiy son). [A, B] kesmani teng n ta kesmaga bo‘ling. [A, B] kesmada ajratilgan barcha nuqtalarni chiqaring.

6.9-masala. n butun soni va sonlar o‘qida 2 ta A, B nuqta berilgan. (A, B haqiqiy son). [A, B] kesmani teng n ta kesmaga bo'ling. [A, B] kesmada ajratilgan barcha

nuqtalar uchun F(X) = 1-sin(X) funksiya qiymatini hisoblang.
6.10-masala. n butun soni berilgan (n>0). Quyidagi ketma - ketlikning dastlabki n ta hadini chiqaruvchi dastur tuzing. $A_0 = 1; A_K=A_{K+1}/K; K = 1,2,...$
6.11-masala. n butun soni berilgan (n>1). Fibonachchi ketma - ketlikning dastlabki n ta hadini chiqaruvchi dastur tuzing. $F_1=1, F_2=1, F_K =F_{K-2}+F_{K-1}$, K=3,4,...
6.12-masala. n butun soni berilgan (n>1). Quyidagi ketma - ketlikning dastlabki n ta hadini chiqaruvchi dastur tuzing. $A_1=1, A_2=2, A_K=(A_{K-2}+2*A_{K-1})/3$, K=3,4,...
6.13-masala. n butun soni berilgan (n > 2). Quyidagi ketma - ketlikning dastlabki n ta hadini chiqaruvchi dastur tuzing. $A_1=1, A_2=2, A_3=3; A_K=A_K+A_{K-2}-2>A_{K-3}$: K=4,5,... ichma - ich ochilgan sikllar
6.14-masala. N butun soni berilgan. Quyidagi yig‘indini chiqaruvchi dastur tuzing. 11+22+...+NN.
6.15-masala. N butun soni berilgan. Quyidagi yig‘indini chiqaruvchi dastur tuzing. 1N+2N-1+...+N1.
6.16-masala. A va B butun soni berilgan (A < B). A va B sonlari orasidagi barcha butun sonlarni chiqaruvchi dastur tuzing. Bunda A soni 1 marta. (A + 1) soni 2 marta chiqariladi va xokazo.
6.17-masala. a va b butun sonlari berilgan (a<b). a va b sonlari orasidagi barcha butun sonlarni (a va b dan tashqari) kamayish tartibida chiqaruvchi va chiqarilgan sonlar sonini aniqlovchi dastur tuzing.
6.18-masala. Bir kilogram konfetning narxi berilgan (haqiqiy son). 1, 2, ..., 10 kg konfetning narxini chiqaruvchi dastur tuzing.
6.19-masala. a va b butun sonlari berilgan (a<b). a dan b gacha bo‘lgan barcha butun sonlar kvadratlarining yig‘indisini chiqaruvchi dastur tuzing.
6.20-masala. n butun soni berilgan (n > 0). Quyidagi yig'indini hisoblovchi dastur tuzing: S=1+1/2 +1/3+...+1/n.

Topshiriq: 2) Quyidagi masalaning PYTHON tilidagi dasturini while takrorlash operatoridan foydalanib tuzing:

7.1-masala. к va n butun sonlari berilgan (n>0). к sonini n marta chiqaruvchi dastur tuzing.
7.2-masala. a va b butun sonlari berilgan (a<b). a va b sonlari orasidagi barcha butun sonlarni (a va b ni ham) chiqaruvchi va chiqarilgan sonlar sonini aniqlovchi dastur tuzing. (a va b ham chiqarilsin).
7.3-masala. Bir kilogram konfetning narxi berilgan (haqiqiy son). 0.1, 0.2, ..., 0.9, 1 kg konfetni narxini chiqaruvchi dastur tuzing.
7.4-masala. Bir kilogram konfetning narxi berilgan (haqiqiy son). 1.2, 1.4, ..., 2 kg konfetni narxini chiqaruvchi dastur tuzing.
7.5-masala. a va b butun sonlari berilgan (a<b). a dan b gacha bo‘lgan barcha butun sonlar yig‘indisini chiqaruvchi dastur tuzing.
7.6-masala. a va b butun sonlari berilgan (a<b). a dan b gacha bo‘lgan barcha butun sonlar ko‘paytmasini chiqaruvchi dastur tuzing.
7.7-masala. n butun soni berilgan (n>0). Quyidagi yig‘indini hisoblovchi dastur tuzing. $S=n^2 +(n+1)^2+(n+2)^2 + ... (2*n)^2$

7.8-masala. n butun soni berilgan (n>0). Quyidagi ko‘paytmani hisoblovchi dastur tuzing. S = 1.1 * 1.2 * 1.3 *... (n ta ko‘paytuvchi)

7.9-masala. n butun soni berilgan (n>0). Quyidagi yig‘indini hisoblovchi dastur tuzing. S=1.1-1.2+1.3-...+$(-1)^n$*1.n ta qo‘shiluvchi, ishoralar almashib keladi. Shart operatoridan foydalanmang)

7.10-masala. n butun soni berilgan (n>0). Shu sonning kvadratini quyidagi formula asosida hisoblovchi dastur tuzing. n^2=1+3+5+...+(2*n -1) har bir qo‘shiluvchidan keyin natijani ekranga chiqarib boring. Natijada ekranda 1 dan n gacha bo‘lgan sonlar kvadrati chiqariladi.

7.11-masala. n butun soni va a haqiqiy soni berilgan (n>0). a ning n-darajasini aniqlovchi dastur tuzing. a^n=a*a*a...a;

7.12-masala. n butun soni va a haqiqiy soni berilgan (n>0). Bir sikldan foydalanib a ning 1 dan n gacha bo‘lgan barcha darajalarini chiqaruvchi dastur tuzing.

7.13-masala. n butun soni va a haqiqiy soni berilgan (n>0). Bir sikldan foydalanib a ning 1 dan n gacha bo'lgan barcha darajalarini chiqaruvchi va yig‘indini hisoblovchi dastur tuzing. 1+a+a^2+a^3+...+a^n.

7.14-masala. n butun soni va a haqiqiy soni berilgan (n>0). Bir sikldan foydalanib a ning 1 dan n gacha bo‘lgan barcha darajalarini chiqaruvchi va yig‘indini hisoblovchi dastur tuzing. 1-a+a^2-a^3+...+$(-1)^n$*a^n.

7.15-masala. n butun soni berilgan (n>0). 1 dan n gacha bo‘lgan sonlar ko‘paytmasini chiqaruvchi dastur tuzing. n! =1*2*...*n. Birdan n gacha bo‘lgan sonlar ko‘paytmasi n faktorial deyiladi.

7.16-masala. n butun soni berilgan (n>0). Bir sikldan foydalangan holda quyidagi yig‘indini hisoblovchi dastur tuzing. 1! + 2! + 3! + ... +n!

7.17-masala. Sportchi birinchi kuni 10 km yugirib boshladi. Keyingi kunlari oldingi kunga nisbatan p foiz ko‘p yugurdi (0<p<50). Sportchining necha kundan keyin jami yugurgan masogasi 200 km dan oshadi? Jami kunlar soni va masofani (butun son) chiqaruvchi dastur tuzing.

7.18-masala. n butun soni berilgan (n>0). Uni bo‘lib butun va qoldiq qismlarini aniqlash orqali, berilgan son raqamlari yig‘indisini va raqamlari sonini chiqaruvchi dastur tuzing.

7.19-masala. n butun soni berilgan (n>1)**.** n sonidan katta bo‘lgan birinchi Fibonachchi sonini aniqlovchi dastur tuzing

7.20-masala. Fibonachchi soni bo‘lgan n butun soni berilgan (n>1). Fibonachchi n sonidan bitta oldingi va bitta keyingi Fibonachchi sonlarini chiqaruvchi dastur tuzing.

V PYTHON DA MASSIVLAR

5.1 MASSIVLAR HAQIDA UMUMIY TUSHUNCHA

Berilganlar massivi tushunchasi: Xotirada ketma-ket (regular) joylashgan bir xil turdagi qiymatlarga massiv deyiladi.

Odatda massivlarga zarurat, katta hajmdagi, lekin cheklangan miqdordagi va tartiblangan qiymatlarni qayta ishlash bilan bog‘liq masalalarni yechishda yuzaga keladi. Faraz qilaylik, talabalar guruhining reyting ballari bilan ishlash masalasi qo‘yilgan. Unda guruhning o‘rtacha reytingini aniqlash, reytinglarni kamayishi bo‘yicha tartiblash, konkret talabaning reytingi haqida ma’lumot berish va boshqa masala ostilarini yechish zarur bo‘lsin. Qayd etilgan masalalarni yechish uchun berilganlarning (reytinglarning) tartiblangan ketma-ketligi zarur bo‘ladi. Bu yerda tartiblanganlik ma’nosi shundaki, ketma-ketlikning har bir qiymati o‘z o‘rniga ega bo‘ladi (birinchi talabaning reytingi massivda birinchi o‘rinda, ikkinchi talabaniki – ikkinchi o‘rinda va hakoza). Berilganlar ketma-ketligini ikki xil usulda hosil qilish mumkin. Birinchi yo‘l – har bir reyting uchun alohida o‘zgaruvchi aniqlash: Reyting1, Reyting2,… ReytingN. Lekin ,guruhdagi talabalar soni yetarlicha katta bo‘lganda, bu o’zgaruvchilar qatnashgan programmani tuzish katta qiyinchiliklarni yuzaga keltiradi. Ikkinchi yo‘l – berilganlar ketma-ketligini yagona nom bilan aniqlab, uning qiymatlariga murojaatni, shu qiymatlarning ketma-ketlikda joylashgan o‘rnining nomeri (indeksi) orqali amalga oshirishdir. Reytinglar ketma-ketligini Reyting deb nomlab, undagi qiymatlariga $Reyting_1$, $Reyting_2$,… $Reyting_N$ ko‘rinishida murojaat qilish mumkin. Odatda berilganlarning bunday ko‘rinishiga massivlar deyiladi. Massivlarni matematikadagi sonlar vektoriga o’xshatish mumkin, chunki vektor ham o‘zining individual nomiga ega va u fiksirlangan miqdordagi bir turdagi qiymatlardan – sonlardan iboratdir.

Demak, massiv – bu fiksirlangan miqdordagi ayrim qiymatlarning (massiv elementlarining) tartiblangan majmuasidir. Barcha elementlar bir xil turda bo‘lishi kerak va bu tur element turi yoki massiv uchun tayanch tur deb nomlanadi. Yuqoridagi keltirilgan misolda Reyting – haqiqiy turdagi vektor deb nomlanadi.

Bu ko‘rinishga xususiy o‘zgaruvchi deyiladi, chunki uning qiymati massivning alohida elementidir, Bizning misolda Reyting massivining alohida komponentalariga Reyting[1], Reyting[2],..., Reyting[N] xususiy o‘zgaruvchilar orqali murojaat qilish mumkin. Boshqacha bu o‘zgaruvchilarni indeksli o‘zgaruvchilar deyiladi.

Umuman olganda indeks sifatida ifoda ishlatilishi mumkin. Ifoda qiymati massiv elementi nomerini aniqlaydi. Ifoda sifatida o‘zgaruvchi ham olinishi mumkinki, o‘zgaruvchining qiymati o‘zgarishi bilan murojaat qilinayotgan massiv elementini aniqlovchi indeks ham o‘zgaradi. Shunday qilib, programmadagi bitta indeksli o‘zgaruvchi orqali massivning barcha elementlarini belgilash (aniqlash)

mumkin. Masalan, Reyting[1] o‘zgaruvchisi orqali I o‘zgaruvchining qiymatiga bog‘liq ravishda Reyting massivining turli (barcha) elementlariga murojaat qilish imkoni mavjud. Shuni qayd qilish kerakki, massiv indeksi sifatida butun son qo‘llaniladi.

Python tilida indeks doimo 0 dan boshlanadi, uning eng katta qiymati massiv e’lonidagi uzunlikdan bittaga kam bo‘ladi.

5.2 PYTHONDA MASSIVLAR

Pythondagi massiv - bu bir xil turdagi ob’ektlarni saqlash uchun ishlatiladigan buyurtma qilingan ma’lumotlar tuzilishi. Funktsional imkoniyatlari jihatidan ular ro’yxatlarga o‘xshashdir, ammo ularning kirish ma’lumotlari turiga, shuningdek o‘lchamlariga nisbatan ba’zi cheklovlar mavjud. Ushbu xususiyatga qaramay, massivlar Python dasturlash tilidagi ma’lumotlar to‘plamlari bilan ishlash uchun juda funktsional vosita hisoblanadi.

Massivlarni yaratish va to‘ldirish. Pythonda yangi qator qo‘shishdan (yaratishdan) oldin, bunday ob’ekt bilan ishlash uchun mas’ul bo‘lgan kutubxonani import qilishingiz kerak. Buning uchun dastur fayliga `from array import *` qator qo‘shilishi kerak. Massivlar bitta doimiy ma’lumotlar turi bilan o‘zaro aloqada bo‘lishga qaratilgan bo‘lib, natijada ularning barcha elementlari bir xil o‘lchamga ega. `array` funktsiyasidan foydalanib biz yangi ma’lumotlar to‘plamini yaratishimiz mumkin.

Massivlarni yaratishning umumiy sintaksisi quyidagicha:

array(massiv_turi, qiymatlar_ro‘yxati)

Quyidagi misol Python massivni qanday to‘ldirish kerakligini ko‘rsatib beradi:

```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
```

Massiv funksiyasi ikkita argumentni oladi, birinchisi - bu yaratilgan massivning turi, ikkinchisi - uning qiymatlarining dastlabki ro‘yxati. Bu yerda massiv elementlarining `'i'` (2 baytli butun) tur. Buning o‘rniga 1 baytli belgi `'c'` (char turi)ni yoki 4 baytli `'f'` float turini kabi boshqa turlardan foydalanishimiz mumkin. Quyidagi jadvalda massiv turlari keltirilgan:

Turning massivda yozilishi	C turi	Python turi	Minimal hajmi baytda
'b'	signed char	int	1
'B'	unsigned char	int	1
'u'	Py_UNICODE	unicode character	2
'h'	signed short	int	2
'H'	unsigned short	int	2
'i'	signed int	int	2
'I'	unsigned int	int	2
'l'	signed long	int	4
'L'	unsigned long	int	4
'q'	signed long long	int	8
'Q'	unsigned long long	int	8
'f'	Float	float	4
'd'	Double	float	8

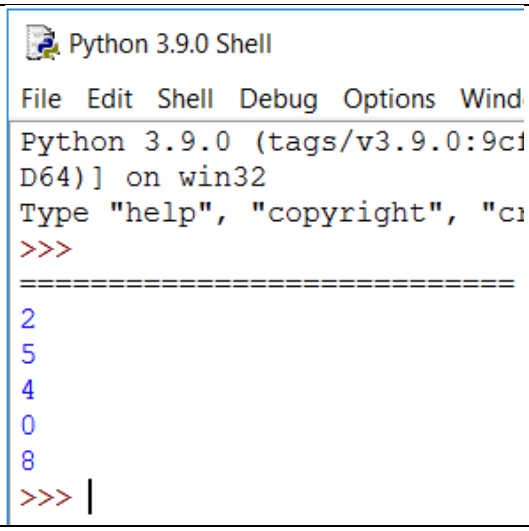
Shuni esda tutish kerakki, massiv faqat bitta turdagi ma'lumotlarni saqlashi mumkin, aks holda dasturni ishga tushirganimizda xatolik beradi va muvaffaqiyatsiz bo'ladi.

Massiv elementiga murojaat qilish. Kvadrat qavs yordamida massiv elementiga murojaat qilishimiz mumkin. Masalan : `massiv[2]`.

Massivlarni ekranga chiqarish. Dasturdagi har qanday ma'lumotlar bilan ishlashda vaqti-vaqti bilan ularni tekshirishga ehtiyoj bor. Bu ularni ekranda aks ettirish orqali osonlikcha amalga oshiriladi. Buni amalga oshirish uchun `print` deb

nomlangan funktsiya yordam beradi. Bu ilgari yaratilgan va to‘ldirilgan qator elementlaridan birini argument sifatida qabul qiladi. Quyidagi misolda for sikl operatori yordamida ma’lumotlar massivining har bir elementi vaqtinchalik identifikator i orqali chiqariladi:

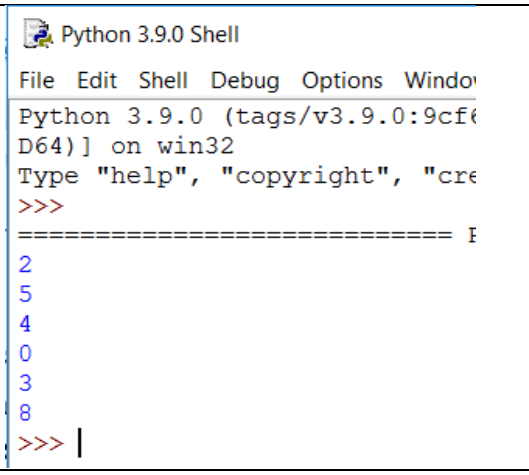
```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
for i in massiv:
    print(i)
```



Yuqoridagi kodning natijasida barcha element qiymatlari bo‘yicha takrorlanadi va ekranga chiqariladi.

Massivga element qo‘shish. Python qatoriga yangi element qo‘shish uchun `insert` metodidan foydalanish kerak. Buning uchun uni avval yaratilgan ob’ekt orqali chaqirish va ikkita qiymatni argument sifatida kiritish kerak. Birinchisi (4) massivdagi yangi elementning indeksiga, ya’ni uni joylashtirish kerak bo'lgan joyga, ikkinchisi (3) qiymatning o‘zi uchun javobgardir.

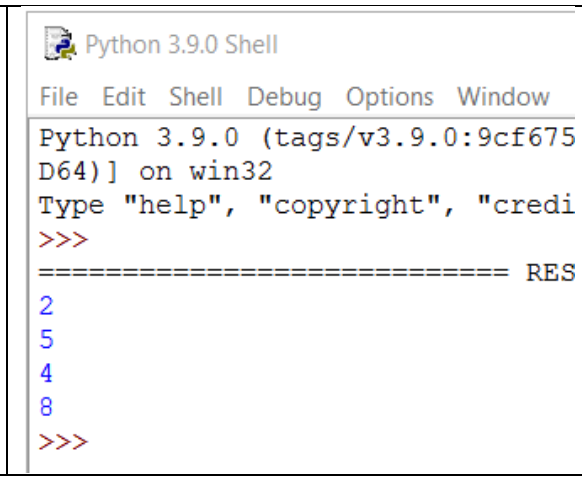
```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
massiv.insert(4, 3)
for i in massiv:
    print(i)
```



Shuni esda tutish kerakki, biz qatorga faqat ilgari yaratilgan ob’ekt tegishli bo‘lgan turdagi ma’lumotlarni qo‘shishimiz mumkin. Bunday operatsiyani bajarishda mavjud bo‘lgan elementlar soni dasturning ehtiyojlariga qarab ko‘payadi.

Elementni o‘chirish. Pythonda pop() metodi yordamida keraksiz elementlarni massivdan olib tashlash mumkin, uning argumenti yacheka indeksi (3). Yangi element qo‘shilgandek bo‘lgani kabi, usulni misolda ko‘rsatilgandek, avval yaratilgan ob’ekt orqali chaqirish kerak.

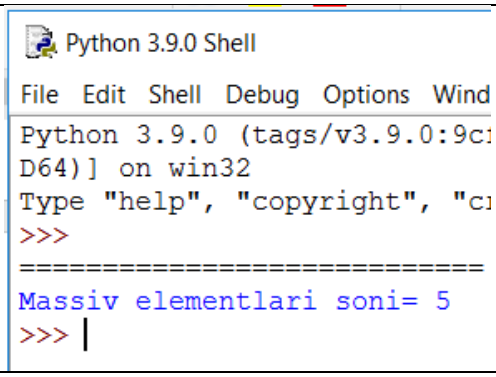
```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
massiv.pop(3)
for i in massiv:
    print(i)
```



Ushbu operatsiyani bajargandan so‘ng, mavjud bo‘lgan xotira katakchalari soni elementlarning joriy soniga to‘g‘ri keladigan qilib massiv tarkibini o‘zgartiradi.

Massiv uzunligini olish. Dastur bajarilishida massivning uzunligi o‘zgarishi mumkinligi sababli, ba’zida uning tarkibidagi elementlarning hozirgi sonini bilish foydalidir. `len()` metodi Pythondagi massivning uzunligini (hajmini) butun son sifatida olish uchun ishlatiladi. Pythonda massiv elementlari sonini ekranga chiqarish uchun `print()` metodidan foydalanamiz:

```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
print("Massiv elementlari soni=",len(massiv))
```

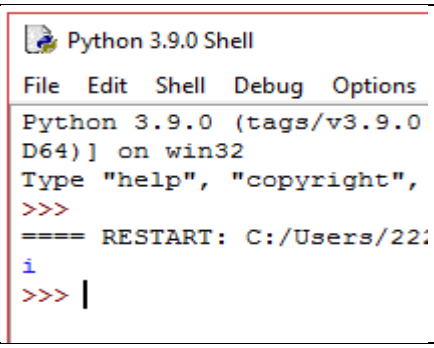


Yuqoridagi dastur kodidan ko‘rinib turibdiki, print() metodi argumenti sifatida len natijasini oladi, bu esa konsolga raqamli qiymatni chiqarishga imkon beradi.

Pythonda massivlar bilan ishlashda qo’llaniladigan funksiyalar va metodlar. Pythonda massivlar ishlashda qo’llaniladigan bir nechta metodlar mavjud bo’lib, ularning eng asosiylari quyida keltirilgan:

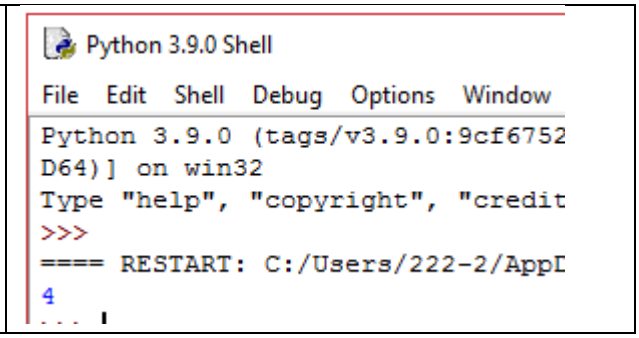
- ***array.typecode*** - Massivning elementlari turini aniqlash uchun ishlatiladi. Agar massivlar bir nechta bo‘lsa array.array(typecode) dan foydalaniladi.

```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
print(massiv.typecode)
```



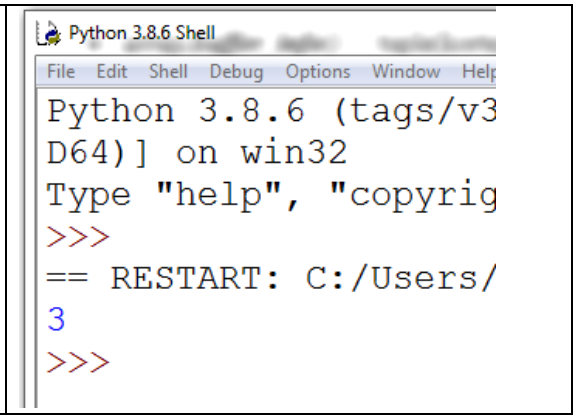
- ***array.itemsize*** - massivdagi bitta elementning baytdagi hajmini hisoblash uchun ishlatiladi.

```
from array import *
massiv = array('i', [2, 5, 4, 0,8])
print(massiv.itemsize)
```



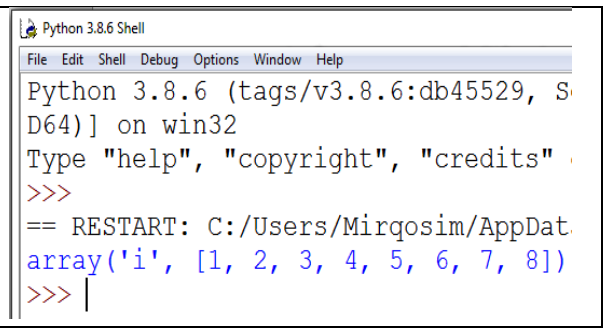
- ***array.count(x)*** - masssivdagi *x* elementlar sonini qiymat sifatida qaytaradi ;

```
from array import *
massiv = array('i', [2,2, 5, 4,4,4, 0,8])
print(massiv.count(4));
```



- ***array.fromlist(ro'yxat)*** – masssivga ro'yxatdagi elementlarni qo'shish uchun ishlatiladi.

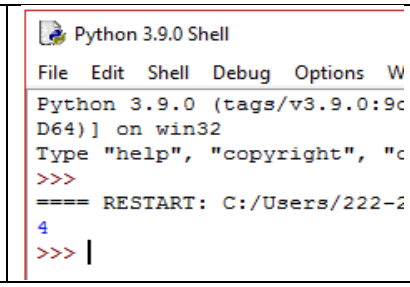
```
from array import *
massiv = array('i',[1,2,3,4,5])
list=[6,7,8]
massiv.fromlist(list)
print(massiv);
```



- ***array.index(x)*** – massivdagi *x* elementining joylashgan indeksini qiymat

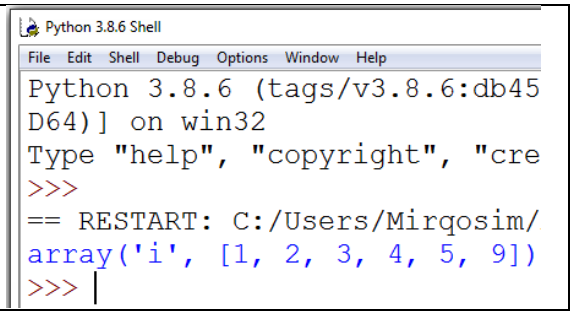
sifatida qaytaradi. Agar bunday element massivda mavjud bo‘lmasa, u holda *ValueError* istisno holati ro‘y beradi;

```
from array import *
massiv = array('i', [2, 5, 4, 0, 8])
print(massiv.index(8))
```



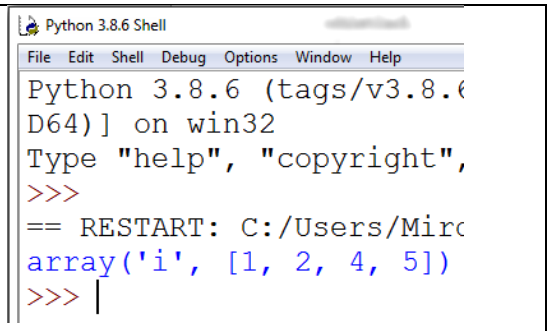
- ***append()*** – metodi massivning oxiriga yangi element qo‘shish uchun foydalaniladi.

```
from array import *
massiv = array('i', [1,2,3,4,5])
massiv.append(9)
print(massiv)
```



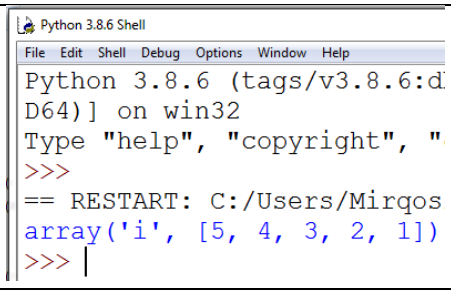
- ***array.remove****(x)* - massivdan *x* elementini o‘chirish. Ushbu metod ro‘yxatdagi birinchi uchragan *x* elementini o‘chiradi. Agar bunday element ro‘yxatda mavjud bo‘lmasa *ValueError* istisno holati ro‘y beradi.

```
from array import *
massiv = array('i', [1,2,3,4,5])
massiv.remove(3)
print(massiv)
```



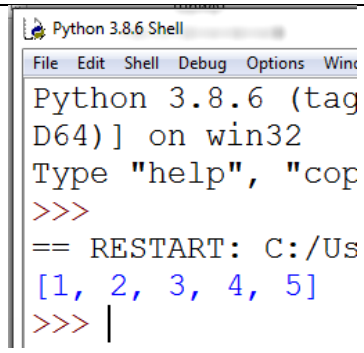
- ***array.reverse()*** - massiv elementlarini teskari tartibda joylashtirish uchun qo‘llaniladi . Bundan tashqari, Python massiv bilan ishlashda qo‘llaniladigan bir nechta standart funksiyalarni ham o‘z ichiga qamrab olgan:

```
from array import *
massiv = array('i', [1,2,3,4,5])
massiv.reverse()
print(massiv)
```



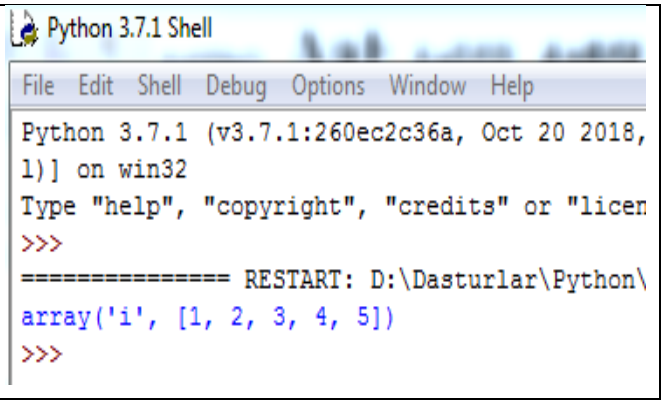
- ***array.tolist()*** -massivni ro‘yxatga aylantirish uchun qo‘llaniladi.

```
from array import *
massiv = array('i', [1,2,3,4,5])
list=massiv.tolist()
print(list)
```



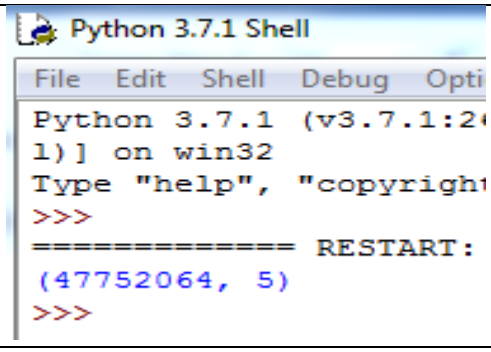
- ***array.tofile(f)*** - massivni ochiq faylga yozish uchun ishlatiladi.
- ***array.fromfile(F,N)*** - fayldan N elementni o'qiydi va ularni massiv oxiriga qo‘shib qo'yadi. Ikkilik o‘qish uchun fayl ochilishi kerak. Agar N dan kam element mavjud bo'lsa, *ValueError* istisnoli tashlanadi, ammo mavjud bo‘lgan elementlar qatorga qo‘shiladi.

```
import array
f=open("array.bin","wb")
massiv=array.array("i",[1,2,3,4,5])
massiv.tofile(f)
f.close()
massiv1=array.array("i")
f=open("array.bin","rb")
massiv1.fromfile(f,len(massiv))
print(massiv1)
```



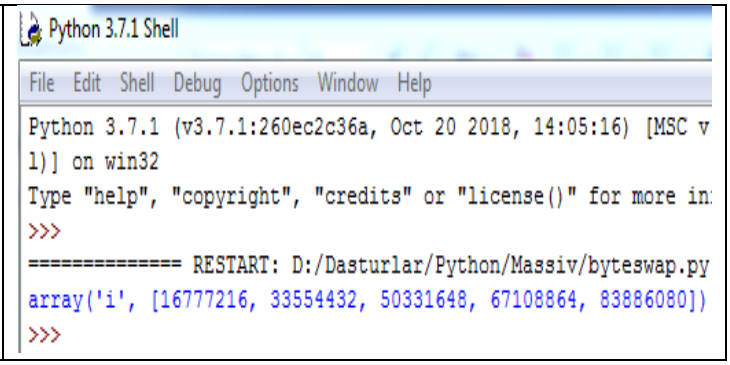
- ***array.buffer_info()*** - tuple(kortej) xotiraning joylashuvi, uzunligini aniqlaydi. Past darajadagi operatsiyalar uchun foydalidir.

```
import array
massiv=array.array('i',[1,2,3,4,5])
print(massiv.buffer_info())
```



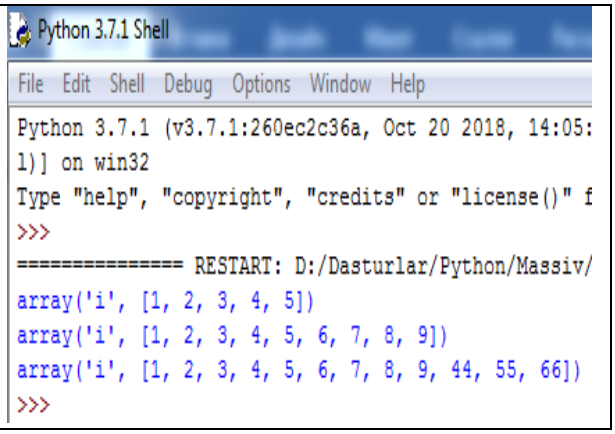
- ***array.byteswap()*** - massivning har bir elementida baytlarning tartibini o‘zgartirish. Chunki boshqa bayt tartibida mashinada yozilgan fayldan ma'lumotlarni o'qishda foydalidir ;

```
from array import array
my_array=array('i',[1,2,3,4,5])
my_array.byteswap()
print(my_array)
```



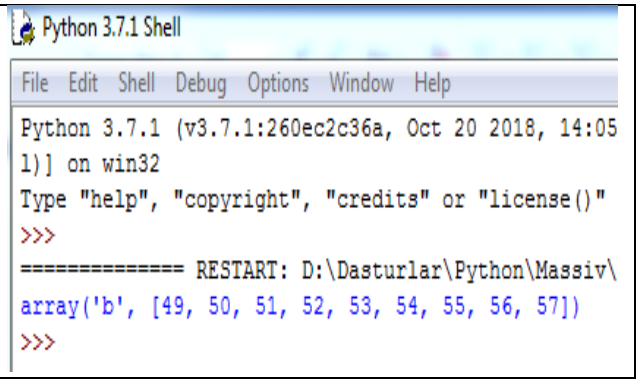
- ***array.extend****(iter)* - massivga ob'ektdan elementlarni qo'shish uchun foydalanadi.

```
from array import array
my_array=array('i')
my_array.extend([1,2,3,4,5])
print(my_array)
my_array.extend(range(6,10))
print(my_array)
my_array.extend(array('i',[44,55,66]))
print(my_array)
```



- ***array.frombytes*** *(b)* - bir qator baytlardan massiv hosil qiladi. Baytlar soni massivdagi bitta element kattaligining ko'paytmasi bo'lishi kerak.

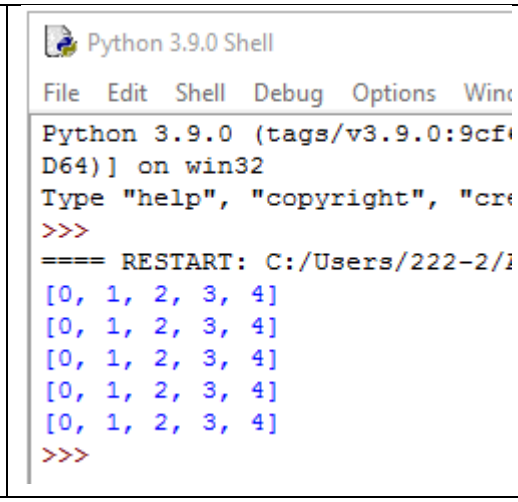
```
from array import array
my_array = array('b')
my_array.frombytes(b'123456789')
print(my_array)
```



5.3 PYTHONDA IKKI VA KO'P O'LCHOVLI MASSIVLAR

Ikki o'lchovli massiv. Ba'zi hollarda oddiy bir o'lchovli massiv ma'lum bir ma'lumot to'plamini to'g'ri ko 'rsatish uchun etarli emas. Python dasturlash tilida ikki o'lchovli va ko'p o'lchovli massivlar mavjud emas, ammo ushbu platformaning asosiy imkoniyatlari ikki o'lchovli ro'yxatni tuzishni osonlashtiradi. Ushbu dizayn elementlari quyidagi misolda ko'rsatilgandek to'ldirilib, ustunlar va qatorlarga joylashtirilgan.

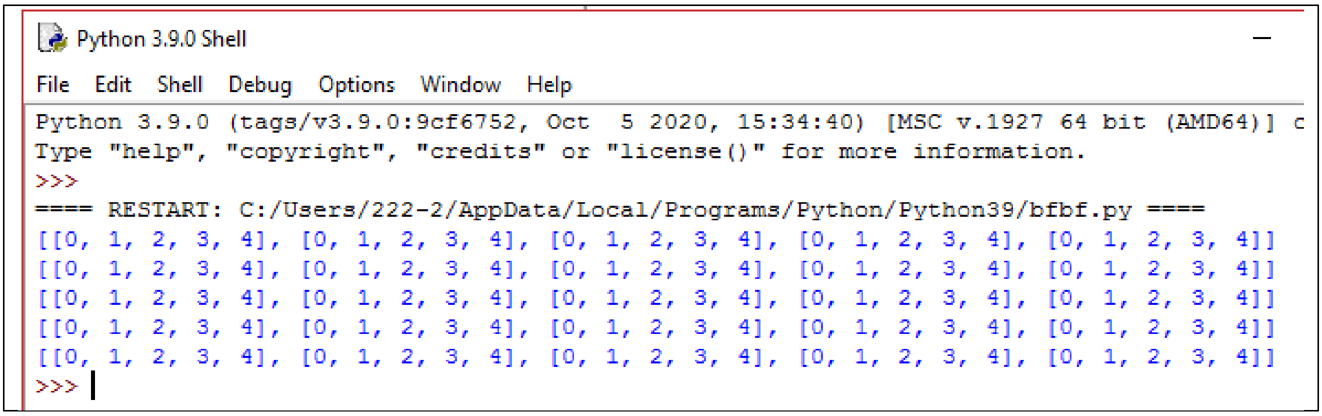
```
from array import *
d1 = []
for j in range(5):
    d2 = []
    for i in range(5):
        d2.append(i)
    d1.append(d2)
for i in d1:
    print(i)
```



Bu yerda biz ikki o‘lchovli ma'lumotlar to'plamini amalga oshirishning asosiy g‘oyasi bitta katta d1 ro‘yxati ichida bir nechta d2 ro‘yxatlarini yaratish ekanligini ko‘rishimiz mumkin. Ikki o‘lchamli 5 × 5 matritsani nol bilan avtomatik to‘ldirish uchun ishlatiladi. Qo‘shish va diapazon usullari ushbu vazifani yengishga yordam beradi, ularning birinchisi ro‘yxatga yangi element qo'shadi (0), ikkinchisi esa uning qiymatini (5) o‘rnatishga imkon beradi. Shuni ta‘kidlash kerakki, for uchun har bir yangi tashqi (j) yoki ichki (i) ro‘yxatlarning joriy elementini ifodalovchi o‘z vaqtinchalik o‘zgaruvchisidan foydalanadi. Ko‘p o‘lchovli ro‘yxatning kerakli katakchasiga uning koordinatalarini to‘rtburchaklar ichida ko'rsatib, satrlar va ustunlarga e'tibor qaratishingiz mumkin: d1 [1] [2].

Ko‘p o‘lchovli massiv. Murakkab ro‘yxat sifatida ko‘rsatilgan ikki o‘lchovli qatorda bo‘lgani kabi, ko‘p o‘lchovli qator ham ro ‘yxat ichida ro‘yxat tarzida amalga oshiriladi. Quyidagi misolda uch o‘lchamli(5x5x5) massiv yaratishni ko‘rib chiqamiz:

```
from array import *
d1 = []
for k in range(5):
    d2 = []
    for j in range(5):
        d3 = []
        for i in range(5):
            d3.append(i)
        d2.append(d3)
    d1.append(d2)
for i in d1:
    print(i)
```



```
Python 3.9.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] c
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/222-2/AppData/Local/Programs/Python/Python39/bfbf.py ====
[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
>>>
```

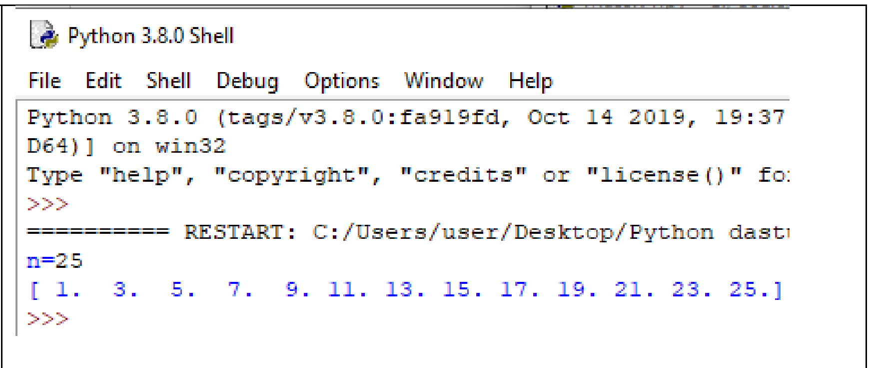
Ikki o‘lchovli massivga o‘xshab, to‘rtburchaklar ichidagi ko‘rsatkichlar yordamida yuqorida qurilgan ob’ekt katakchasiga murojaat qilishimiz mumkin.
Masalan, d1 [4] [2] [3].

Massivlar odatda Python dasturlash tilidagi bir xil turdagi ma’lumotlar to’plamlari bilan o‘zaro aloqada bo'lish uchun ishlatiladi. Platformaning standart kutubxonasi sizga tegishli funktsiyalar yordamida uning tarkibini boshqarish qobiliyatini ta’minlaydigan bunday tuzilma bilan samarali ishlashga imkon beradi. Bundan tashqari, Python sathlar soniga cheklovlarsiz ro‘yxatlarning ko‘p o‘lchovli namoyishini qo‘llab-quvvatlaydi.

5.4. PYTHON DA MASSIVLAR TADBIQI

8.1-masala. n natural soni berilgan. Dastlabki n ta toq sondan tashkil topgan massivni hosil qiling va elementlarini chiqaring.

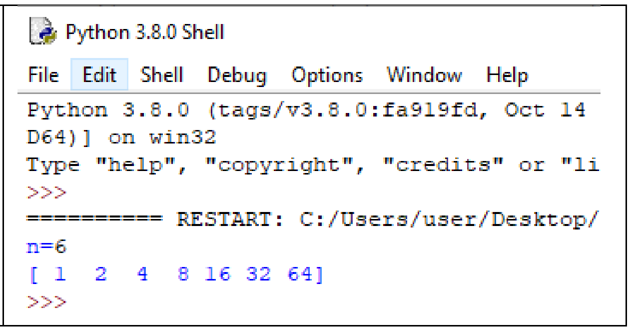
```
import numpy as np
n=int(input('n='))
toq =
np.array(range(1,n+1,2),
float)
print(toq)
```



```
Python 3.8.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37
D64)] on win32
Type "help", "copyright", "credits" or "license()" fo:
>>>
========== RESTART: C:/Users/user/Desktop/Python dast
n=25
[ 1.  3.  5.  7.  9. 11. 13. 15. 17. 19. 21. 23. 25.]
>>>
```

8.2-masala. n natural soni berilgan. 2 soninig dastlabki n ta darajasidan tashkil topgan massivni hosil qiling va elementlarini chiqaring. (1, 2, 4, 8)

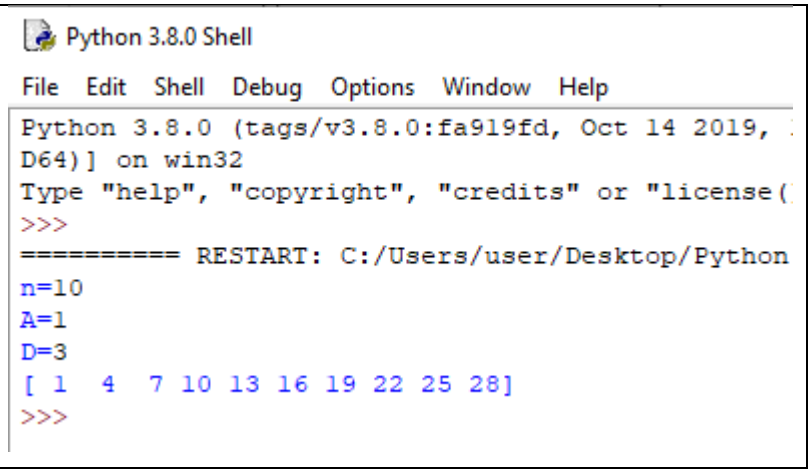
```
import numpy as np
n=int(input('n='))
a = np.array(range(n+1), int)
for i in range(n+1):
   a[i]=2**i
print(a)
```



```
Python 3.8.0 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14
D64)] on win32
Type "help", "copyright", "credits" or "li
>>>
========== RESTART: C:/Users/user/Desktop/
n=6
[ 1  2  4  8 16 32 64]
>>>
```

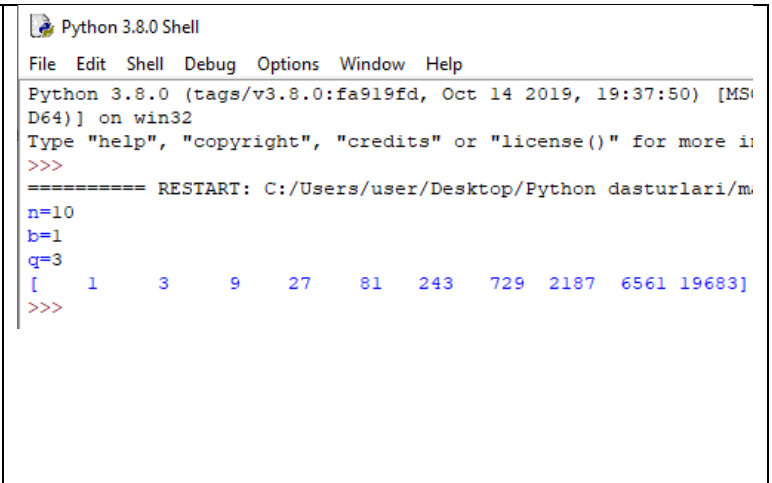
8.3-masala. n natural soni va arifmetik progressiyaning dastlabki hadi A va ayirmasi D berilgan. Arifmetik progressiyaning dastlabki n ta hadidan tashkil topgan massivni hosil qiling va elementlarini chiqaring.

```
import numpy as np
n=int(input('n='))
A=int(input('A='))
D=int(input('D='))
a = np.array(range(n), int)
for i in range(0,n):
    a[i]=A+D*i
print(a)
```



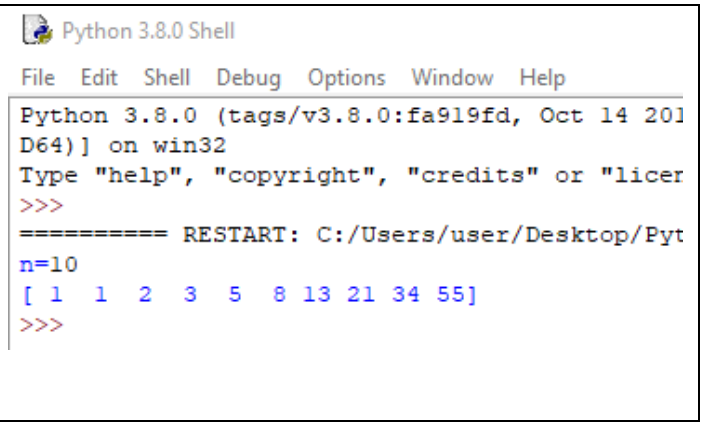
8.4-masala. n natural soni va geometrik progressiyaning dastlabki hadi b va maxraji q berilgan. Geometrik progressiyaning dastlabki n ta hadidan tashkil topgan massivni hosil qiling va elementlarini chiqaring.

```
import numpy as np
n=int(input('n='))
b=int(input('b='))
q=int(input('q='))
massiv = np.array(range(n), int)
massiv[0]=b
for i in range(1,n):
    massiv[i]=massiv[i-1]*q
print(massiv)
```



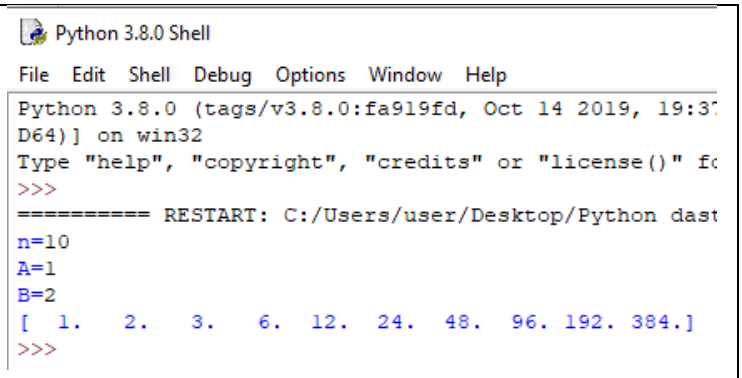
8.5-masala. n natural soni berilgan. Dastlabki n ta Fibonachchi sonlaridan tashkil topgan massivni hosil qiling va elementlarini chiqaring.
F[0] = 1; F[1] = 1; F[k] = F[k-1] + F[k-2]; k=2, 3, 4, ...

```
import numpy as np
n=int(input('n='))
massiv = np.array(range(n), int)
massiv[0]=1
massiv[1]=1
for i in range(2,n):
    massiv[i]=massiv[i-1]+massiv[i-2]
print(massiv)
```



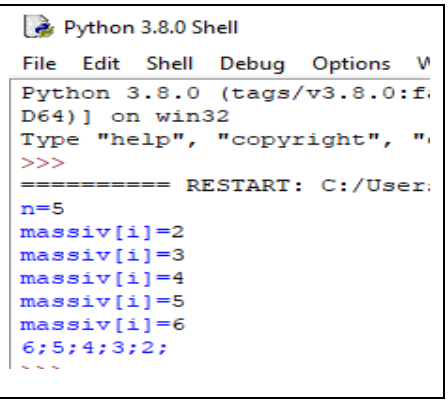
8.6-masala. n natural soni va A, B butun sonlari berilgan (n > 2). a[0] = A; a[1] = B; boshqa elementlari o'zidan oldingi barcha elementlari yig'indisiga teng bo'lgan massivni hosil qiling va elementlarini chiqaring.

```
import numpy as np
n=int(input('n='))
A=int(input('A='))
B=int(input('B='))
massiv = np.array(range(n), float)
massiv[0]=A
massiv[1]=B
s=A+B
for i in range(2,n):
    massiv[i]=s  s+=massiv[i]
print(massiv)
```



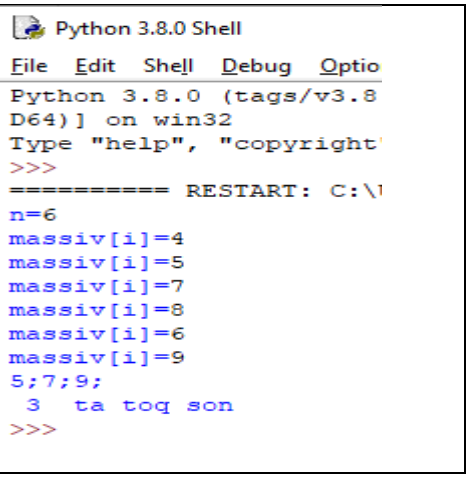
8.7-masala. n ta elementdan tashkil topgan massiv berilgan. uning elementlarini teskari tartibda chiqaruvchi programma tuzilsin.

```
import numpy as np
n=int(input('n='))
massiv = np.array(range(n), int)
for i in range(0,n):
    massiv[i]=int(input('massiv[i]='))
for i in range(n-1,-1,-1):
    print(massiv[i],end=';')
```



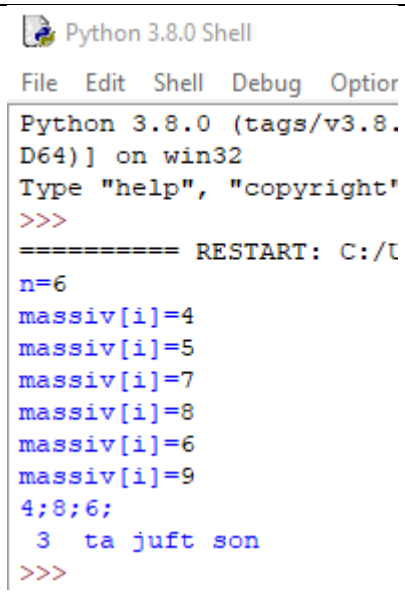
8.8-masala. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan toqlarini chiqaruvchi va ularning sonini chiqaruvchi programma tuzilsin. Massiv elementlar: 4 5 7 8 6 9 Natija: 5 7 9 toqlar soni = 3

```
import numpy as np
n=int(input('n='))
massiv = np.array(range(n), int)
for i in range(0,n):
    massiv[i]=int(input('massiv[i]='))
k=0
for i in range(0,n):
    if massiv[i]%2==1: print(massiv[i],end=';')
        k+=1 print("\n",k," ta toq son")
```



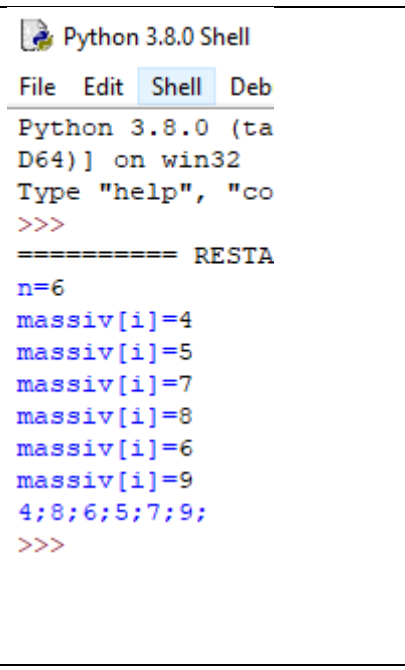
8.9-masala. n ta elementdan tashkil topgan massiv berilgan. Massiv elementlari orasidan juftlarini chiqaruvchi va ularning sonini chiqaruvchi programma tuzilsin.
Massiv elementlar: 4 5 7 8 6 9 Natija: 4 8 6 juftlar soni = 3

```
import numpy as np
n=int(input('n='))
massiv = np.array(range(n), int)
for i in range(0,n):
    massiv[i]=int(input('massiv[i]='))
k=0
for i in range(0,n):
    if massiv[i]%2==0:
        print(massiv[i],end=';')
        k+=1
print("\n",k," ta juft son")
```



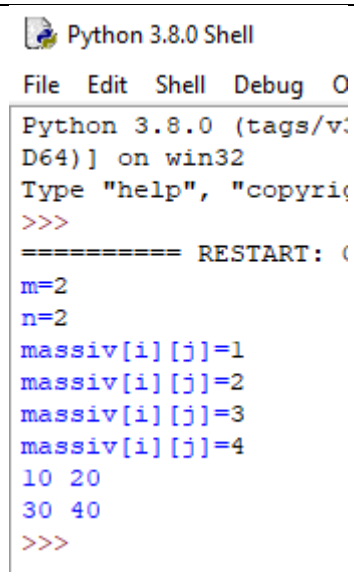
8.10-masala. n ta elementdan tashkil topgan massiv berilgan. Dastlab massiv elementlari orasidan juftlarini chiqaruvchi, keyin massiv elementlari orasidan toqlarini chiqaruvchi programma tuzilsin.
Massiv elementlar: 4 5 7 8 6 9 Natija: 4 8 6 5 7 9

```
import numpy as np
n=int(input('n='))
massiv = np.array(range(n), int)
for i in range(0,n):
    massiv[i]=int(input('massiv[i]='))
for i in range(0,n):
    if massiv[i]%2==0:
        print(massiv[i],end=';')
for i in range(0,n):
    if massiv[i]%2==1:
        print(massiv[i],end=';')
```



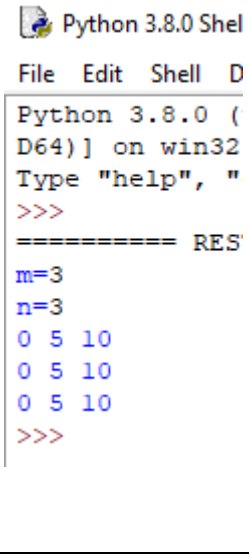
8.11-masala. m va n butun musbat sonlari berilgan. m x n o'lchamli matritsani shunday hosil qilingki, uning har bir i - satri elementlari 10 * i ga teng bo'lsin. (i =0, 1, m -1)

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)]
for j in range(n)])
for i in range(m):
    for j in range(n):


massiv[i][j]=int(input('massiv[i][j]='))
for i in range(m):
    for j in range(n):
        massiv[i][j]*=10
        print(massiv[i][j],end=' ')
    print('')
```



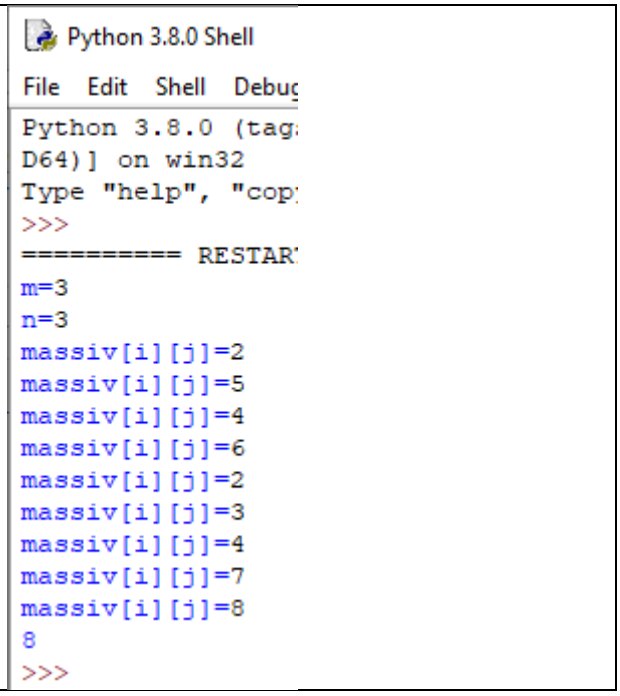
8.12-masala. m va n butun musbat sonlari berilgan. m x n o'lchamli matritsani shunday hosil qilingki, uning har bir j - ustuni elementlari 5 * j ga teng bo'lsin.
(j =0, 1, ..., n -1)

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in
range(n)])
for i in range(m):
    for j in range(n):
        massiv[i][j]=5*j
        print(massiv[i][j],end=' ')
    print('')
```



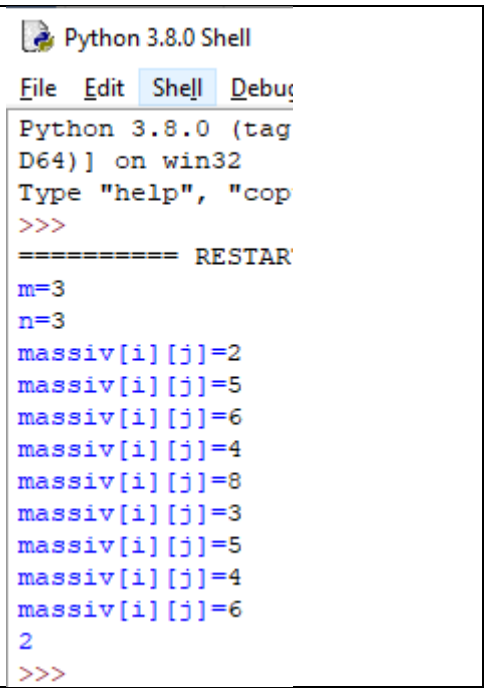
8.13-masala. m va n butun musbat sonlari berilgan. m x n o'lchamli matritsani shunday hosil qilingki, undagi eng katta elementini toping.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)]
for j in range(n)])
for i in range(m):
    for j in range(n):

massiv[i][j]=int(input('massiv[i][j]='))
print(massiv.max())
```



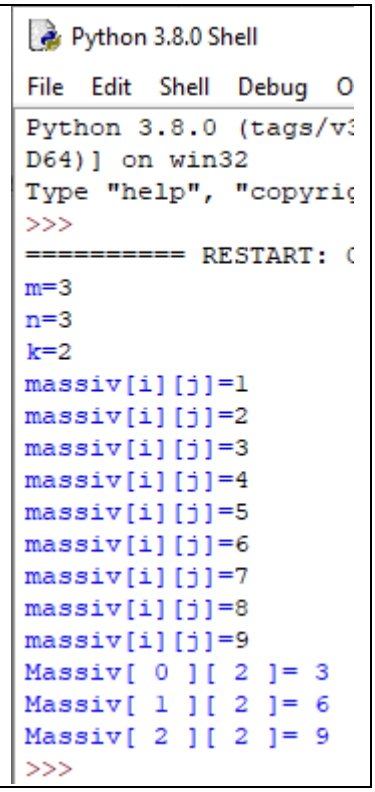
8.14-masala. m va n butun musbat sonlari berilgan. m x n o'lchamli matritsani shunday hosil qilingki, undagi eng kichik elementini toping.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in
range(n)])
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
print(massiv.min())
```



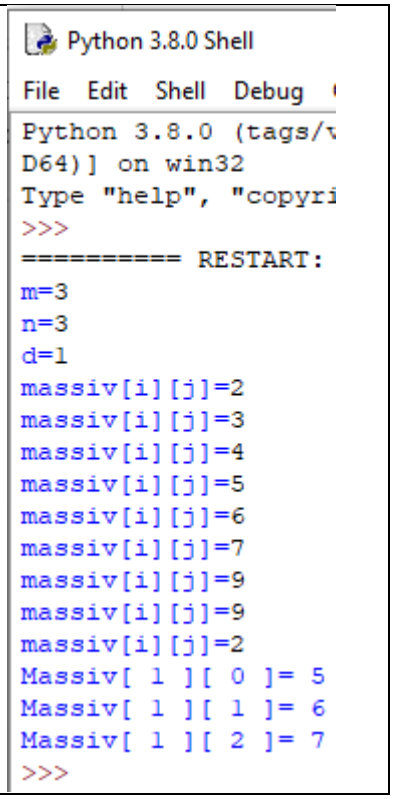
8.15-masala. m x n o‘lchamli massiv berilgan. Shu massivning k-satr ustunini toping.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in range(n)])
k=int(input('k='))
massiv=[[0 for i in range(m)] for j in range(n)]
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
for i in range(m):
    print("Massiv[",i,"][",k,"]=",massiv[i][k])
```



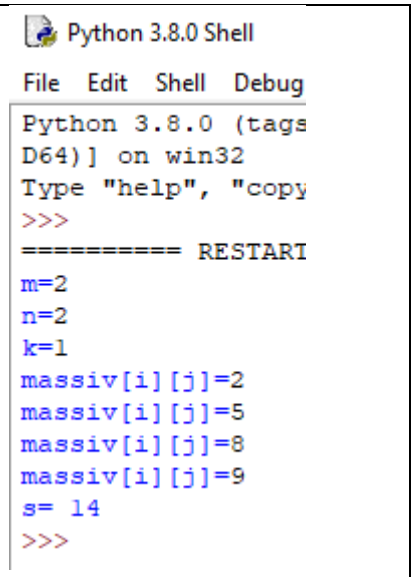
8.16-masala. m x n o‘lchamli massiv berilgan. Shu massivning d-satr qatorini toping.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in
range(n)])
d=int(input('d='))
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
for j in range(n):
    print("Massiv[",d,"][",j,"]=",massiv[d][j])
```



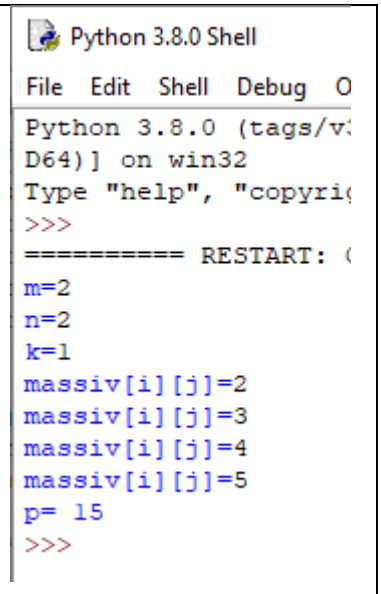
8.17-masala. m x n o‘lchamli matritsa va κ soni berilgan (0 <=k < m). Matritsaning k – ustun elementlari yig‘indisini chiqaruvchi programma tuzilsin.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in
range(n)])
k=int(input('k='))
s=0
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
if k>=0 and k<m:
   for i in range(m):
       s+=massiv[i][k]
else:
    print("Shart bajarilmadi!")
print('s=',s)
```



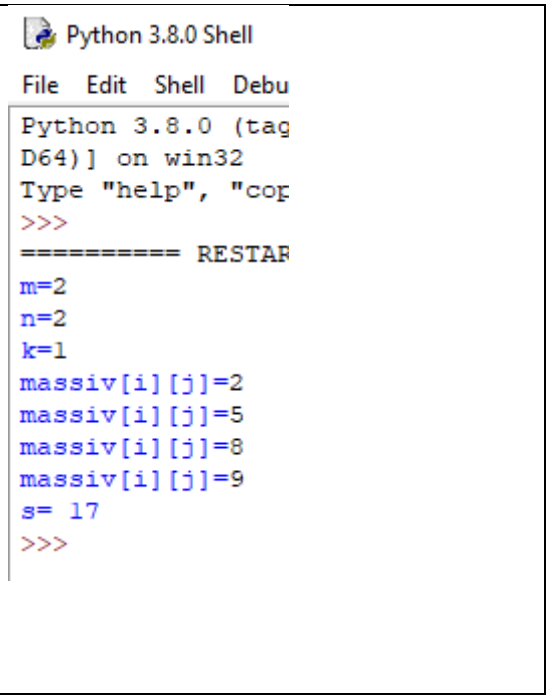
8.18-masala. m x n o'lchamli matritsa va κ soni berilgan (0 <= k < m). Matritsaning k – ustun elementlari ko‘paytmasini chiqaruvchi programma tuzilsin.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in range(n)])
k=int(input('k='))
p=1
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
if k>=0 and k<m:
   for i in range(m):       p*=massiv[i][k]
else:     print("Shart bajarilmadi!")
print('p=',p)
```



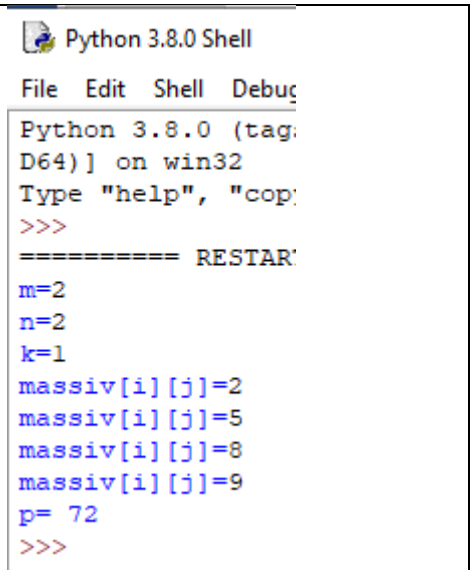
8.19-masala. m x n o'lchamli matritsa va κ soni berilgan (0 <= k < m). Matritsaning k – satri elementlari yig‘indisini chiqaruvchi programma tuzilsin.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j
in range(n)])
k=int(input('k='))
s=0
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
if k>=0 and k<m:
   for i in range(m):
       s+=massiv[k][i]
else:
    print("Shart bajarilmadi!")
print('s=',s)
```



8.20-masala. m x n o'lchamli matritsa va κ soni berilgan (0 <= κ < m). Matritsaning k – satri elementlari ko‘paytmasini chiqaruvchi programma tuzilsin.

```
import numpy as np
m=int(input('m='))
n=int(input('n='))
massiv=np.array([[0 for i in range(m)] for j in
range(n)])
k=int(input('k='))
p=1
for i in range(m):
    for j in range(n):
        massiv[i][j]=int(input('massiv[i][j]='))
if k>=0 and k<m:
   for i in range(m):
       p*=massiv[k][i]
else:
    print("Shart bajarilmadi!")
print('p=',p)
```



5.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Topshiriq: 1) Quyidagi massivlarni PYTHON dasturlash tilida tuzing:
8.1-masala. *n* o‘lchamli *a* massiv va $k(1 \le k \le n)$ butun soni berilgan. Massiv elementlari shart operatoridan foydalanmasdan quyidagi tartibda chop etilsin:

$a_k, a_{k-1}, a_{k-2}, \dots a_1$.

8.2-masala. *n* o‘lchamli *a* massiv berilgan(*n*-juft son). (indekslari o‘sish tartibida) Juft indeksdagi elementlari chiqarilsin. $a_2, a_4, \dots, a_n$. Shart operatoridan foydalanilmasin.

8.3-masala. *n* o‘lchamli *a* massiv berilgan(*n*-toq son). Massivning toq indeksida turgan elementlari indekslarini kamayish tartibida tartiblab chiqarilsin. $a_n, a_{n-2}, a_{n-4}, \dots a_1$ shart operatoridan foydalanilmasin.

8.4-masala. *n* o‘lchamli *a* massiv berilgan. Avval massivning juft indeksli elementlari(indekslarini o‘sish tartibida) keyin toq indeksli elementlari (indekslarini o‘sish tartibida) chiqarilsin: $a_2, a_4, a_6, \dots a_1, a_3, a_5 \dots$ Shart operatoridan foydalanilmasin.

8.5-masala. *n* o‘lchamli *a* massiv berilgan. Avval toq indeksdagi elementlar, keyin juft indeksdagi elementlar kamayish tartibida chop etilsin.

8.6-masala. *n* o‘lchamli *a* massiv berilgan. Uning elementlari quyidagi tartibda chiqarilsin: $a_1, a_n, a_2, a_{n-1}, a_3, a_{n-2}, \dots$

8.7-masala. . *n* o‘lchamli *a* massiv berilgan. Uning elementlari quyidagi tartibda chiqarilsin: $a_1, a_2, a_n, a_{n-1}, a_3, a_4, a_{n-2}, a_{n-3}, \dots \dots$(n-juft son).

8.8-masala. *n* o‘lchamli nol bo‘lmagan butun tipli *a* massiv berilgan. Uning $a_k < a_n$ tengsizlikni qanoatlantiradigan birinchi a_k elementining qiymati chiqarilsin.

8.9-masala. *n* o‘lchamli butun tipli *a* massiv berilgan. Uning $a_1 < a_{k<} a_n$ qo‘sh tengsizlikni qanoatlantiradigan oxirgi a_k elementining tartib nomeri chiqarilsin.

8.10-masala. .*n* o‘lchamli massiv hamda *k* va *l* butun sonlari berilgan$(1 \le k \le l \le n)$. *k*-indeksdan *l*-indeksgacha bo‘lgan massiv elementlarining yig‘indisi topilsin.

8.11-masala. O‘lchamli matritsa berilgan. Uning elementlari quyidagi tartibda chop etilsin: 1-satr elementlarini chapdan o‘ngga, 2-satr elementlarini o‘ngdan chapga, 3-satr elementlarini chapdan o‘ngga, 4-satr elementlarini o‘ngdan chapga va hokazo.

8.12-masala. O‘lchamli matritsa berilgan. Uning elementlari quyidagi tartibda chop etilsin: 1-ustun elementlarini tepadan pastga, 2-ustun elementlarini pastdan tepaga va hokazo.

8.13-masala. O‘lchamli *a* kvadrat matritsa berilgan. Uning boshlang‘ich elementi a_{11} hisoblanadi. Uning elementlari quyidagi ko‘rinishda chiqarilsin: barcha 1-satrdagi elementlarini; oxirgi ustun elementlarini, ($a_{1,m}$ elementdan tashqarisini); 2-satrdagi

ekranga chiqmagan elementlarini, oxiridan oldingi ustundagi chop etilmagan elementlarini va hokazo; eng oxirida a_{m1} elementi chop etilsin.

8.14-masala. $m \times m$ o‘lchamli a kvadrat matritsa berilgan. Uning boshlang‘ich elementi a_{11} hisoblanadi. Uning elementlari quyidagi ko‘rinishda chiqarilsin: 1-ustundagi barcha elementlar; oxirgi satrdagi chop etilmagan elementlar (1-elementdan tashqari); 2-ustundagi qolgan elementlar, oxiridan oldingi satrdagi qolgan element va hokazo; hamda eng oxirida $a_{1,m}$ element chop etilsin.

8.15-masala. m - tartibli a kvadrat matritsa berilgan(m-toq son). Element $a_{1,1}$ dan boshlanadi. Matritsa elementlari soat strelkasi bo‘yicha spiralsimon ko‘rinishda joylashtirilib, matritsa chop etilsin : 1-satr, oxirgi ustunning qolgan elementlari yuqoridan pastga qarab, oxirgi satrning qolgan elementlari o‘ngdan chapga qarab, 1-ustunning qolgan elementlari pastdan yuqoriga qarab, 2-satrning qolgan elementlari chapdan o‘ngga qarab va hokazo. Oxirida markazdagi element chop etilsin.

8.16-masala. m - tartibli a kvadrat matritsa berilgan(m-toq son). Element $a_{1,1}$ dan boshlanadi. Matritsa elementlari soat strelkasiga teskari tartibda spiralsimon ko‘rinishda chop etilsin: 1-ustun, oxirgi satrning qolgan elementlari, oxirgi ustunning qolgan elementlarini quyidan yuqoriga qarab, 1-satrning qolgan elementlarini o‘ngdan chapga qarab, 2-ustunning qolgan elementlarini yuqoridan pastga qarab va hokazo. Eng oxirida markazdagi element chop etilsin.

8.17-masala. $m \times n$ o‘lchamli matritsa va $k(1 \le k \le m)$ butun son berilgan. Berilgan matritsaning k-satridagi elementlarining yig‘indisi va ko‘paytmasi chop etilsin.

8.18-masala. $m \times n$ o‘lchamli matritsa va $k(1 \le k \le m)$ butun son berilgan. Berilgan matritsaning k-ustunidagi elementlarining yig‘indisi va ko‘paytmasi chop etilsin.

8.19-masala. $m \times n$ o‘lchamli matritsa berilgan. Uning har bir satri uchun yig‘indilar hisoblansin.

8.20-masala. $m \times n$ o‘lchamli matritsa berilgan. Uning har bir ustuni uchun yig‘indilar hisoblansin.

VI BOB. PYTHON DA FUNKSIYALAR

6.1. FUNKSIYALARNI TA’RIFLASH VA CHAQIRISH

Funksiyalar parametrlar, ya’ni funksiyaga berilishi mumkin bo‘lgan qiymatlar qabul qila oladi va ular ustuda biror amal bajarishi mumkin. Bu parametrlar o’zgaruvchilarga o’xshaydi. Faqat ulardan farqi bu o’zgaruvchilarning qiymati funksiyani chaqirish vaqtida o’rnatiladi. Funksiya ish boshlagan vaqtda bularga qiymat biriktirilgan bo‘ladi.

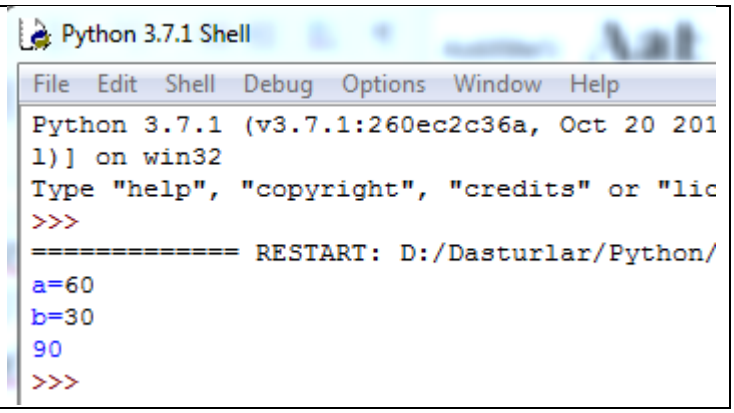
Parametrlar funksiya aniqlanayotgan vaqtda qavs ishida vergul bilan ajratilgan holda ko‘rsatiladi. Ularga qiymatni funksiyani chaqirganimizda biriktiramiz. Ushbu atamalarga e’tibor bering: funksiya e’lon qilinayotgan vaqtda ko‘rsatilgan nomlar **parametrlar**, funksiyani chaqirayotganimizda unga berilgan qiymatlar esa **argumentlar** deyiladi.

Funksiya – bu ko‘p marta ishlatiladigan dastur bo‘lagi. Funksiyalar ma’lum buyruqlar blokini ko‘rsatilgan nom bilan saqlash va shu blokni dasturning istalgan joyida, istalgan miqdorda bajarish imkonini beradi. Biz oldingi darslarimizda **len** va **range** Python funksiyalari bilan tanishgan edik.

Funksiyalar **def** zahira so‘zi orqali aniqlanadi. Bu so‘zdan so‘ng funksiya **nomi**, undan so‘ng qavs va shu qavs ichida bir necha o‘zgaruvchilarni ko‘rsatish mumkin bo‘ladi va oxirida ikki nuqta (:) yoziladi. Shulardan so‘ng funksiyani tashkil qiluvchi buyruqlar bloki yoziladi. Quyidagi misolda buning oson ekanligini ko‘rish mumkin.

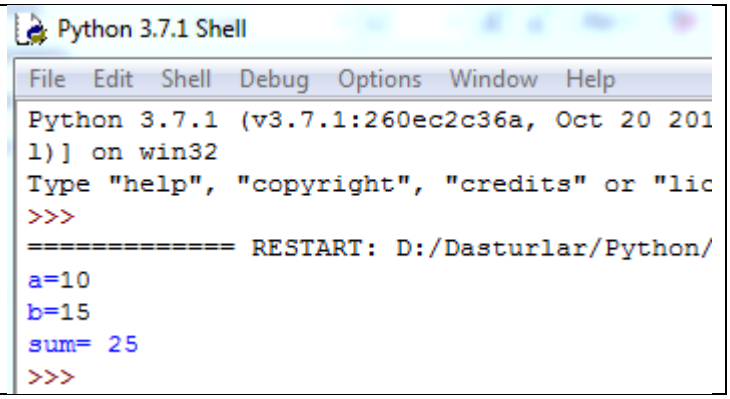
Sodda funksiyaga misol.

```
def ikkita_sum():
    a=int(input('a='));
    b=int(input('b='));
    sum=a+b;
    return sum;
print(ikkita_sum());
```



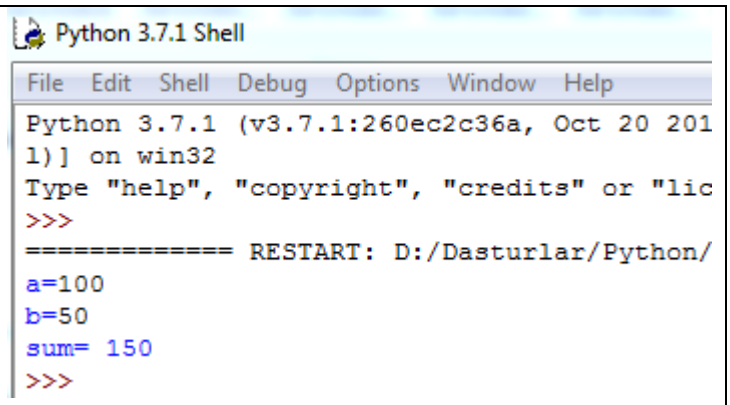
Bu misolda ikki son yig‘indisini xisoblovchi funksiya ko‘rsatilgan. Bu funksiya birorta argument qabul qilmaydi, yig‘indini hisoblab natijani chiqaradi. Shundan so‘ng **print** operatori tanasida natijani chiqarish uchun chaqiriladi. Bu funksiyani shunday o‘zgartiramizki, qiymatni qaytarmasdan, chiqarsin. Buning uchun **print** operatorini funksiya tanasiga kiritish etarli:

```
def ikkita_sum():
    a=int(input('a='));
    b=int(input('b='));
    sum=a+b;
    print('sum=',sum);
ikkita_sum();
```



a va **b** o‘zgaruvchilarni argument sifatida e’lon qilishimiz mumkin, bu xolda funksiya tanasida ularni tariflash talab etilmaydi.

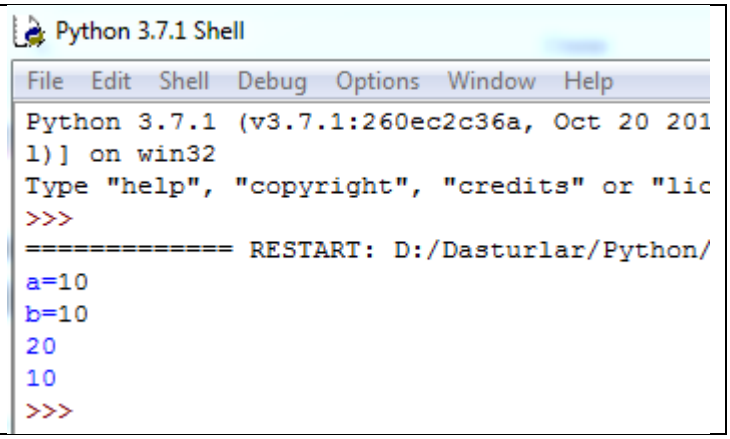
```
def ikkita_sum(a,b):

    sum=a+b;

    print('sum=',sum);

a=int(input('a='));

b=int(input('b='));

ikkita_sum(a,b);
```



Argument orqali uzatilgan qiymatni o‘z ichiga oluvchi o‘zgaruvchi, funksiya **parametri** deyiladi.

Ko‘rilgan misollarda funksiya argumenti qiymati bo‘yicha uzatiladi, ya’ni argumentlar funksiya ichida o‘zgarib, ular funksiya tashqarisidagi qiymatlarga ta’sir qilmaydi:

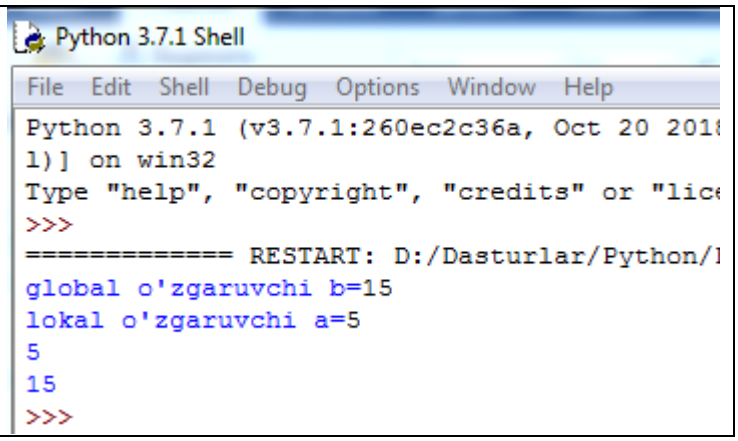
```
def ikkita_sum(a):
    sum=a+10;
    return sum;
a=int(input('a='));
b=int(input('b='));
print(ikkita_sum(a));
print(b);
```



6.2. O‘ZGARUVCHILARNING KO‘RINISH SOHASI

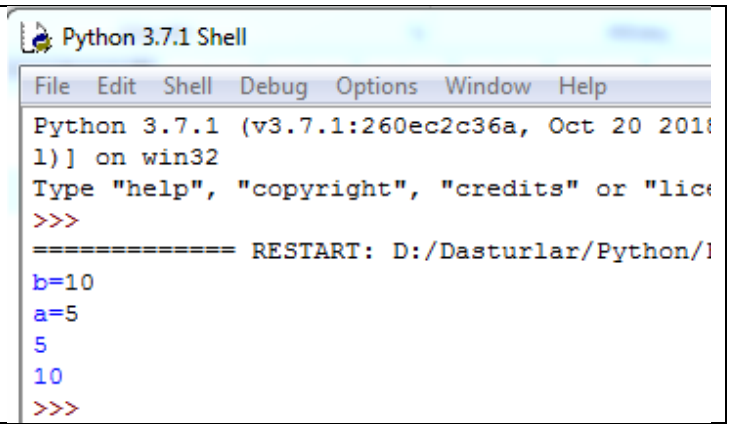
O‘zgaruvchilar funksiyalarda lokal ko‘rinish sohasiga ega. Bu shuni bildiradiki, hatto lokal va tashqi o‘zgaruvchilar bir xil nomga ega bo‘lsa ham, lokal o‘zgaruvchi o‘zgarishi tashqi o‘zgaruvchiga ta’sir qilmaydi.

```
def get_sum():
    a=int(input("lokal o'zgaruvchi
a="));
    print(a);
b=int(input("global o'zgaruvchi
b="));
get_sum();
print(b);
```



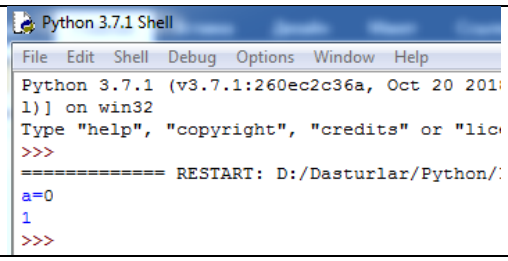
Lokal o‘zgaruvchini global qilish mumkin, agar uning nomi oldidan **global** kalit so‘zi ko‘rsatilsa. Agar tashqi o‘zgaruvchi **global** sifatida e’lon qilingan bo‘lsa, unga ixtiyoriy funksiyadan murojaat qilish mumkin:

```
def get_sum():
    global a;
    a=int(input('a='));
    print(a);
b=int(input('b='));
get_sum();
print(b);
```



O‘zgaruvchi xayot davri deb u mavjud bo‘lgan dastur bajarilish intervali tushuniladi. Lokal o‘zgaruvchilar ko‘rinish sohasi funksiya bo‘lgani uchun, ularning xayot davri ular ta’riflangan funksiya bajarilish vaqti bilan belgilanadi. Bu shuni bildiradiki, har xil funksiyalarda bir - biridan mustaqil ravishda bir xil nomli o‘zgaruvchilar ishlatilishi mumkin. Lokal o‘zgaruvchi har gal funksiya chaqirilganda yangidan initsializatsiya qilinadi, shuning uchun quyidagi misolda keltirilgan sanovchi funksiyaning qaytaruvchi qiymati har gal 1 ga teng bo‘ladi:

```
def a():
    a=int(input('a='));
    return a+1;
print(a());
```

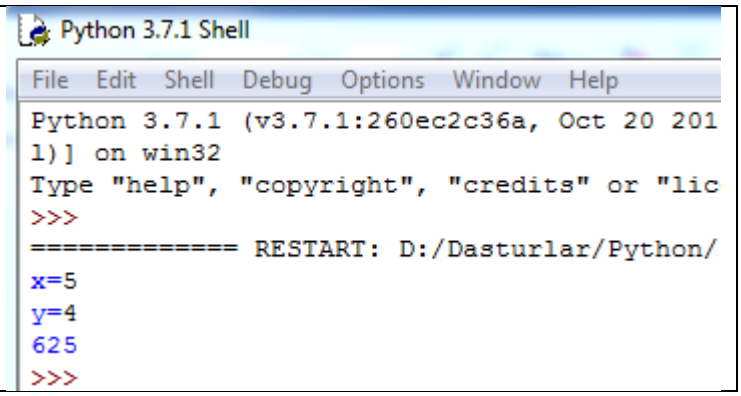


6.3. REKURSIYA TUSHUNCHASI

Rekursiya deb shunday konstruktsiyaga aytiladiki, funksiya o‘zini - o‘zi chaqiradi. To‘g‘ri va nisbiy rekursiya bir – biridan farqlanadi. Funksiya to‘g‘ri rekursiv deyiladi, agar tanasida o‘ziga murojaat mavjud bo‘lsa. Funksiya boshqa funksiyani chaqirsa va bu funksiya o‘z navbatida birinchi funksiyani chaqirsa, bunday funksiya nisbiy rekursiv funsiya deyiladi.

Rekursiyani qo‘llashga klassik misollar sifatida darajaga oshirish va son faktorialini hisoblash keltirish mumkin. Bu misollar rekursiyani tushuntirish qulay bo‘lgani uchun klassik hisoblanadi, lekin ular iteratsion usullarga nisbatan afzallikka ega emas.

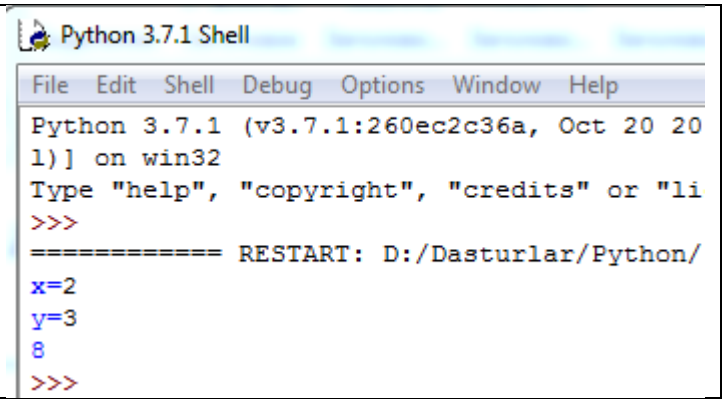
```
x=int(input('x='));
y=int(input('y='));
def degree(x,y):
    if(y):
        return x*degree(x,y-1);
    return 1;
print(degree(x,y));
```



Bu misol quyidagi qoidaga asoslangan $\mathbf{x^y}$ ekvivalent $\mathbf{x*x^{(y-1)}}$. Bu kodda $\mathbf{2^4}$ hisoblash masalasi, $\mathbf{2*2^3}$ hisoblashga keltiriladi. So‘ng $\mathbf{2*2^3}$ ni hisoblash $\mathbf{2*2^2}$ ni hisoblashga keltiriladi, toki ko‘rsatkich nolga teng bo‘lmaguncha.

Bu misolning iteratsion varianti quyidagi ko‘rinishga ega:

```
x=int(input('x='));
y=int(input('y='));
def degree(x,y):
    result=1;
    while y>0:
        y-=1;
        result*=x;
    return result;
print(degree(x,y));
```

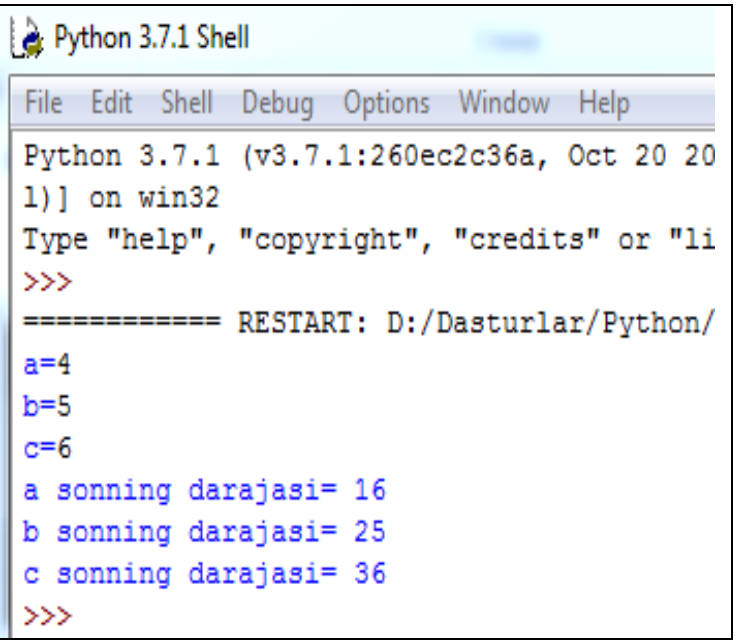


Bu kodni tushunish osonligidan tashqari, u samaraliroqdir, chunki siklni bajarish funksiyani chaqirishga nisbatan tez bajariladi.

6.4. FUNKSIYA TADBIQI

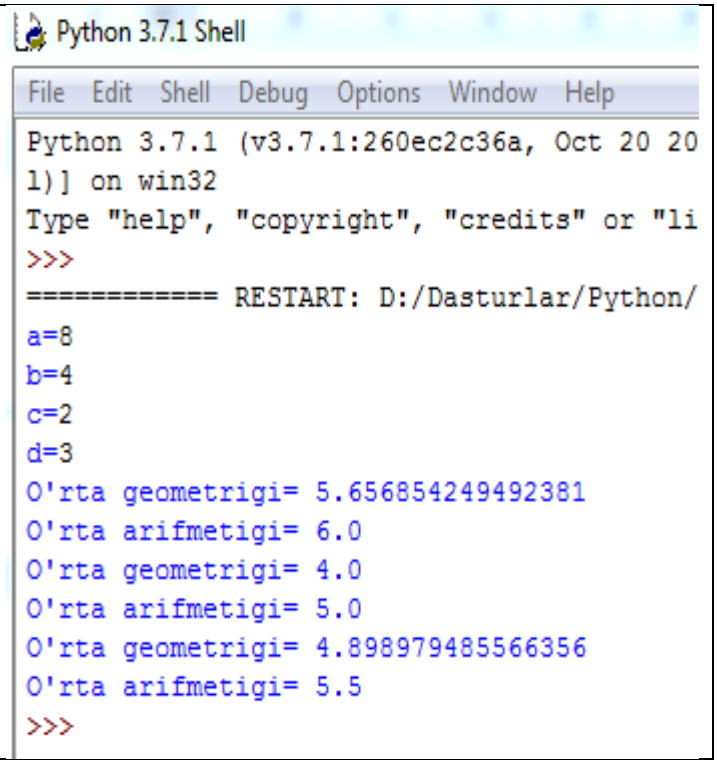
9.1-masala. Ihtiyoriy sonning darajasini hisoblovchi Daraja2 nomli funksiya hosil qiling. Daraja2 funksiyasi orqali a, b, c sonlarining darajasini hisoblovchi dastur tuzing.

```
def Daraja2(a):
    a=a*a;
    return a;
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
print("a sonning darajasi=",Daraja2(a));
print("b sonning darajasi=",Daraja2(b));
print("c sonning darajasi=",Daraja2(c));
```



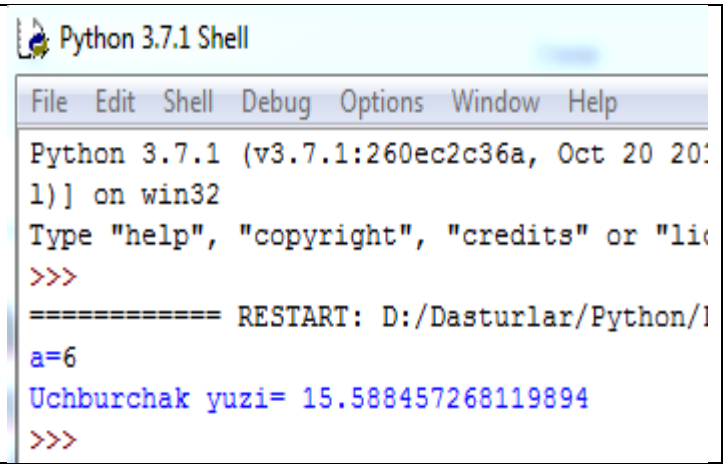
9.2-masala. Ikkita sonning o‘rta arifmetigi va geometrigini hisoblovchi o‘rta_arifmetigi_geometrigi nomli funksiya hosil qiling. o‘rta_arifmetigi_geometrigi funksiyasi orqali a, b, c, d sonlaridan (a, b), (a, c), (a, d) juftliklarining o‘rta arifmetigi va geometrigini hisoblovchi dastur tuzing.

```
import math;
def orta_arifmetigi_geometrigi(a,b):
    p=math.sqrt(a*b);
    s=(a+b)/2;
    print("O'rta geometrigi=",p);
    print("O'rta arifmetigi=",s);
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
d=int(input('d='));
orta_arifmetigi_geometrigi(a,b);
orta_arifmetigi_geometrigi(a,c);
orta_arifmetigi_geometrigi(a,d);
```



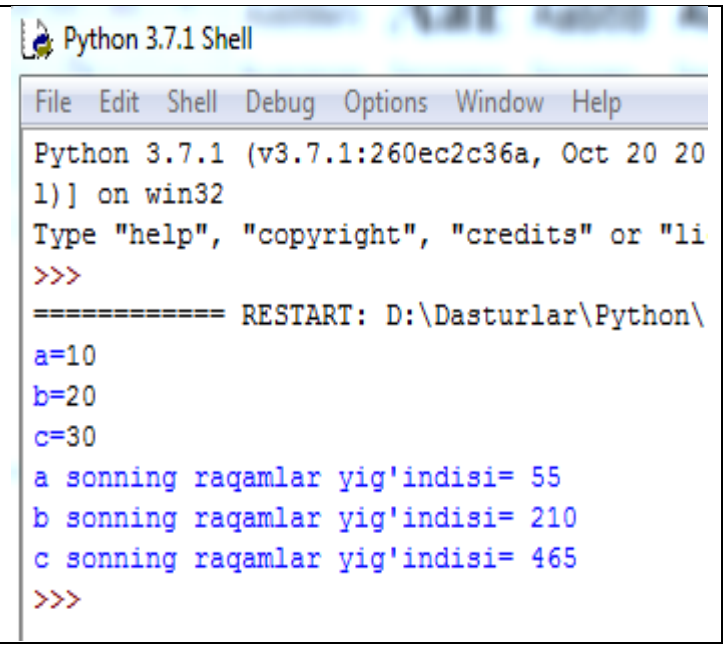
9.3-masala. Teng tomonli uchburchakning yuzasini hisoblovchi uchburchak_yuzi nomli funksiya hosil qiling. Uchburchak_yuzi funksiyasi orqali uchta teng tomonli uchburchakning yuzini hisoblovchi dastur tuzing.

```
import math;
def Uchburchak_yuzi(a):
    s=a*a*math.sqrt(3)/4;
    return s;
a=int(input('a='));
print("Uchburchak
yuzi=",Uchburchak_yuzi(a));
```



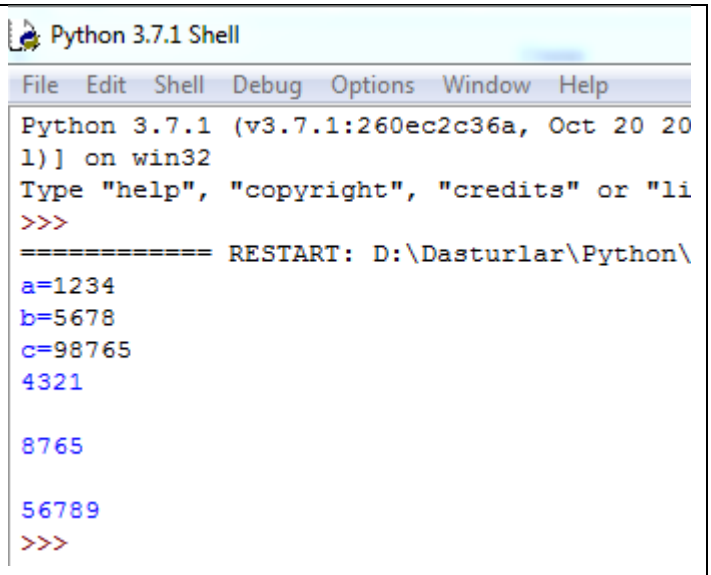
9.4-masala. Natural sonning raqamlar yig‘indisini hisoblovchi raqamlar_yig‘indisi nomli funksiya hosil qiling. Bu funksiya orqali a, b, c sonlarining yig‘indisini hisoblovchi dastur tuzing.

```
def raqamlar_yigindisi(n):
    s=0;
    for i in range(1,n+1):
        s=s+i;
    return s;
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
print("a sonning raqamlar
yig'indisi=",raqamlar_yigindisi(a));
print("b sonning raqamlar
yig'indisi=",raqamlar_yigindisi(b));
print("c sonning raqamlar
yig'indisi=",raqamlar_yigindisi(c));
```



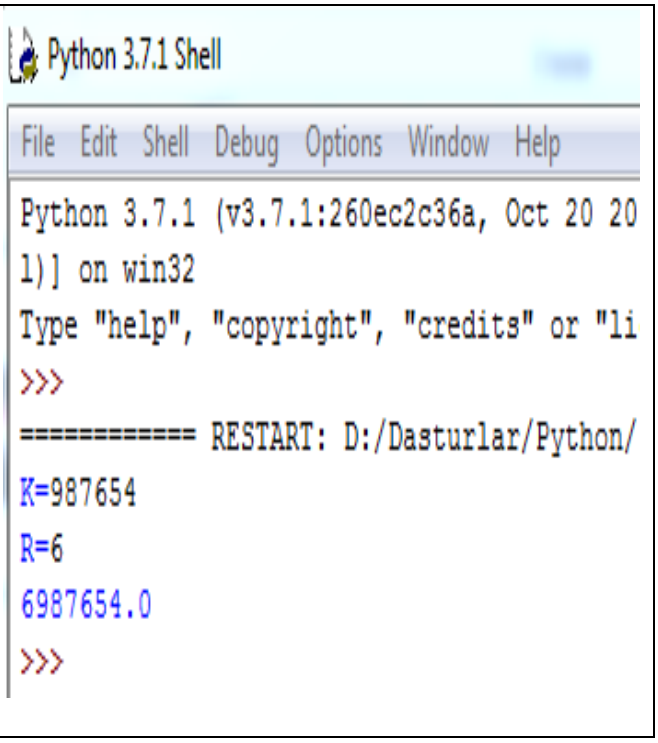
9.5-masala. Butun musbat sonning raqamlarini teskari tartibda chiqaruvchi teskari_tartibda nomli funksiya hosil qiling. Bu funksiya orqali a, b, c sonlarining raqamlarini teskari tartibda chiqaruvchi dastur tuzing.

```
import math;
def teskari_tartibda(n):
    while n>0:
        i=n%10;
        n=math.floor(n/10);
        print(i,end='');
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
teskari_tartibda(a);
print('\n');
teskari_tartibda(b);
print('\n');
teskari_tartibda(c);
```



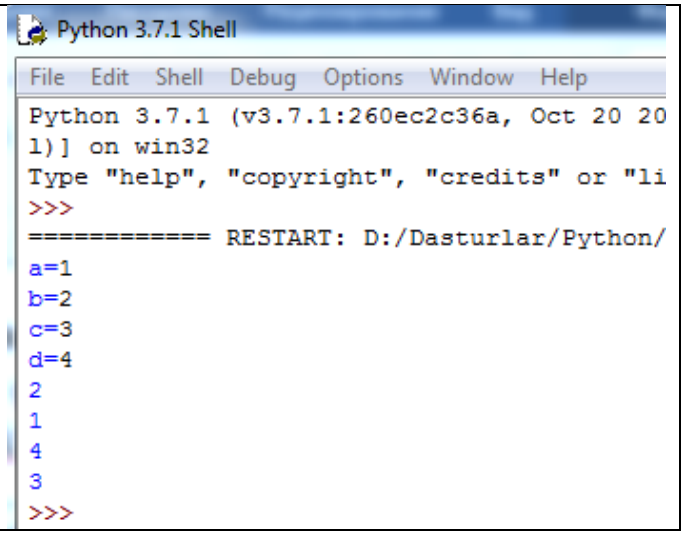
9.6-masala. Kiritilgan K butun musbat sonining chap tarafiga (boshiga) R raqamini (1<=R<=9) qo‘shuvchi RQo‘shish nomli funksiya hosil qiling.

```
import math;
def Rqoshish(son,raqam):
    Raqamlar_soni=0;
    x=son;
    while x>0:
        x=math.floor(x/10);
        Raqamlar_soni+=1;
son= son+raqam*
math.pow(10,Raqamlar_soni);
    print(son);
k=int(input('K='));
r=int(input('R='));
if 1<=r and r<=9:
    Rqoshish(k,r);
else:
    print("R ni 1 va 9 oraliqda
kiriting!");
```



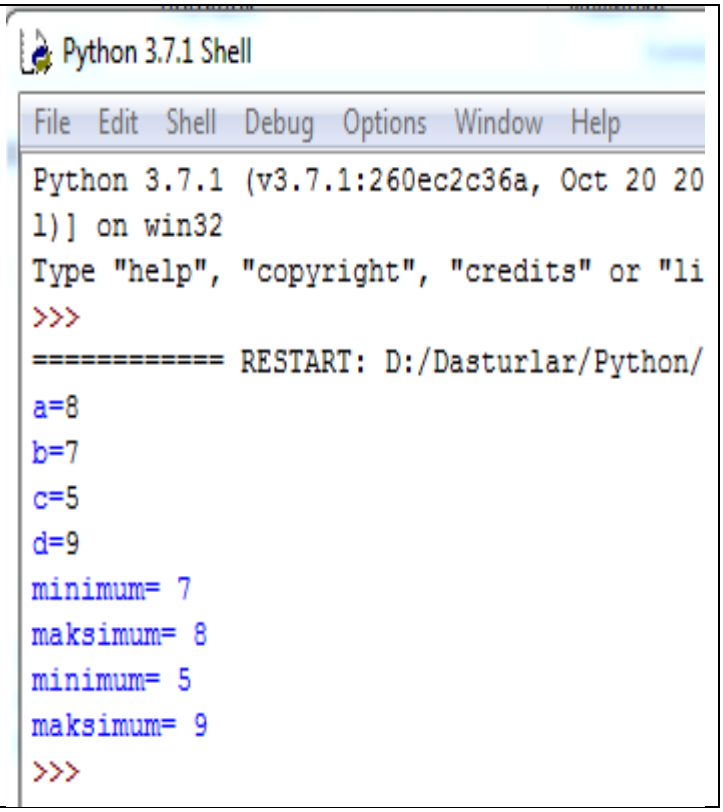
9.7-masala. Ikkita sonning qiymatini almashtiruvchi almashtirish nomli funksiya hosil qiling. Almashtirish funksiyasi orqali A. B, C, D sonlaridan (A, B), (D, C) juftliklarining qiymatlarini almashtiruvchi dastur tuzing.

```
def almashtirish(x,k):
    y=x;
    x=k;
    k=y;
    print(x);
    print(k);
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
d=int(input('d='));
almashtirish(a,b);
almashtirish(c,d);
```



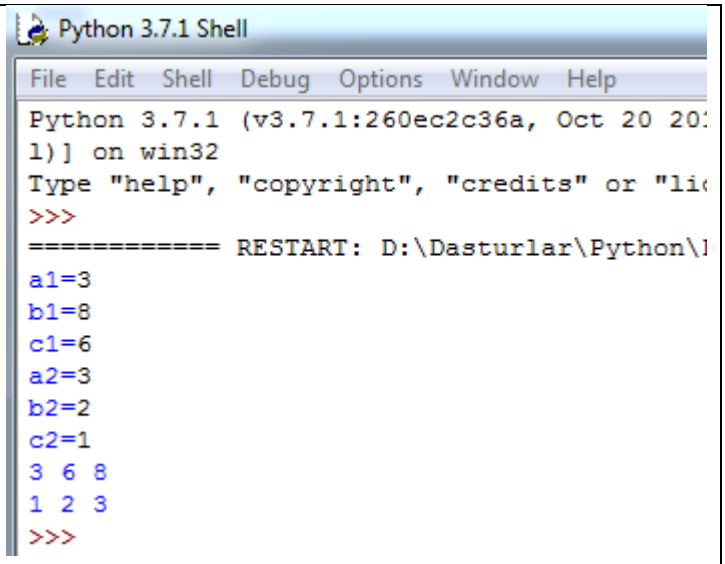
9.8-masala. X va Y sonlaridan kichigini X ga va kattasini Y ga yozuvchi Minmax(X,Y) funksiyasini hosil qiling. Minmax funksiyasini 4 marta chaqirish orqali a. b, c, d butun sonlaridan kattasini va kichigini aniqlovchi dastur tuzing.

```
def Minmax(x,y):
    if x>y:
        max=x;
        min=y;
    else:
        max=y;
        min=x;
    print("minimum=",min);
    print("maksimum=",max);
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
d=int(input('d='));
Minmax(a,b);
Minmax(c,d);
```



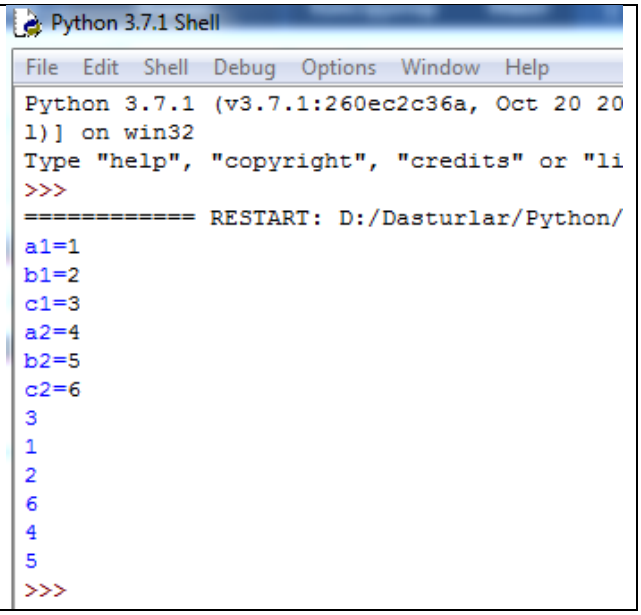
9.9-masala. a, b, c sonlarini o‘sish tartibida joylashtiruvchi O‘sish_tartibi(a, b, c) funksiyasini hosil qiling. Ya‘ni a, в, c sonlari qiymatlarini shunday almashtiringki, natijada a ning qiymati eng kichik va c ning qiymati eng katta bo‘lsin. Bu funksiya orqali (a1, b1, c1) va (a2, b2, c2) sonlarini tartiblang.

```
def usish_tartibi(a,b,c):
    if a>=b and a>=c:
        if b>=c:
            print(c,b,a);
        else:
            print(b,c,a);
    elif b>a and b>c:
        if a>c:
            print(c,a,b);
        else:
            print(a,c,b);
    elif c>a and c>b:
        if a>b:
            print(b,a,c);
        else:
            print(a,b,c);
a1=int(input('a1='));
b1=int(input('b1='));
c1=int(input('c1='));
a2=int(input('a2='));
b2=int(input('b2='));
c2=int(input('c2='));
usish_tartibi(a1,b1,c1);
usish_tartibi(a2,b2,c2);
```



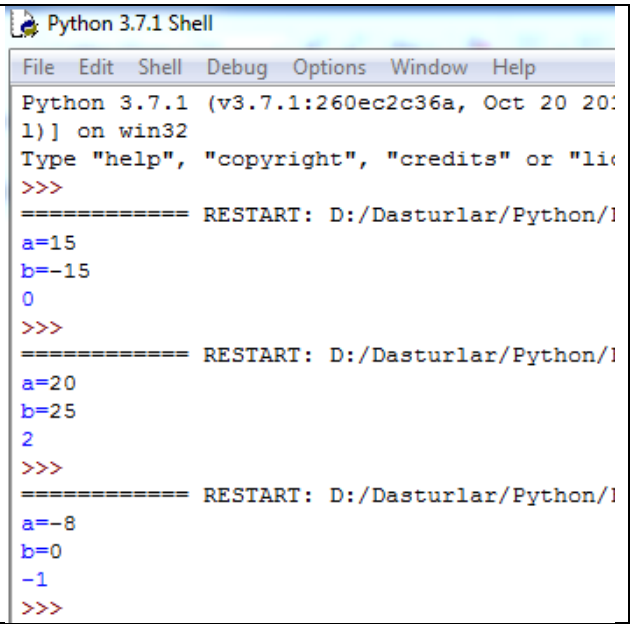
9.10-masala. O‘ngga siklik siljishni amalga oshiruvchi O‘nga_siljish(A, B, C) funksiyasini hosil qiling. Ya‘ni A ning qiymati B ga, B ning qiymati C ga, C ning qiymati A ga o‘tib qolsin. Bu funksiya orqali (A1, B1, C1) va (A2. B2, C2) sonlarini siljiting.

```
def ungga_siljish(a,b,c):
    k=c;  c=b;  b=a;  a=k;
    print(a);
    print(b);
    print(c);
a1=int(input('a1='));
b1=int(input('b1='));
c1=int(input('c1='));
a2=int(input('a2='));
b2=int(input('b2='));
c2=int(input('c2='));
ungga_siljish(a1,b1,c1);
ungga_siljish(a2,b2,c2);
```



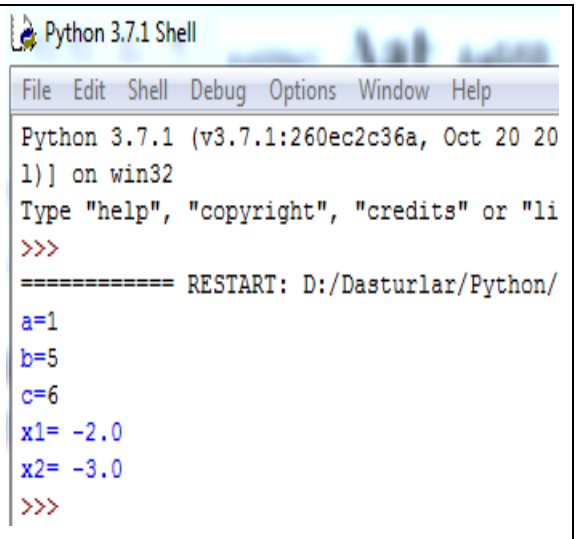
9.11-masala. Haqiqiy sonning ishorasini aniqlovchi ishora nomli funksiya hosil qiling. Funksiya argumenti noldan kichik bo‘lsa -1; noldan katta bo‘lsa 1; nolga teng bo‘lsa 0 qiymat qaytarsin. Haqiqiy a va b sonlari uchun ishora(a) + ishora(b) ifodasi hisoblansin.

```
def ishora(n):
    if n<0:
        n=-1;
    elif n>0:
        n=1;
    else:
        n=0;
    return n;
a=float(input('a='));
b=float(input('b='));
print(ishora(a)+ishora(b));
```



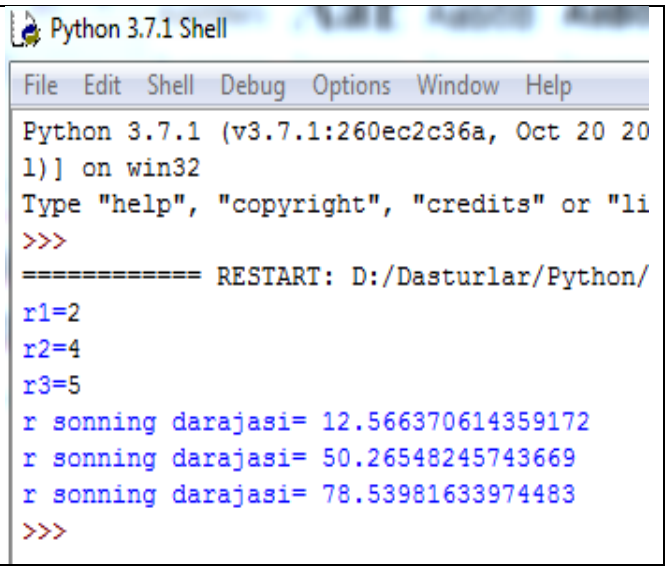
9.12-masala. Kvadrat tenglamaning ildizlari sonini aniqlovchi funksiya hosil qiling. $a * x^2 + b * x + c = 0$ ko‘rinishidagi tenglama kvadrat tenglama deyiladi. (a noldan farqli son)

```
import math;
def kvadrat_tenglama(a,b,c):
    if a!=0:
        d=math.pow(b,2)-4*a*c;
        if d>0:
            x1=(-b+math.sqrt(d))/(2*a);
            x2=(-b-math.sqrt(d))/(2*a);
            print('x1=',x1,'\nx2=',x2);
        elif d==0:
            x=-b/(2*a);
            print('x=',x);
        else:
            print('Yechim mavjud emas!');
    else:
        print("a sonni 0 dan farqli son kiriting!");
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
kvadrat_tenglama(a,b,c);
```



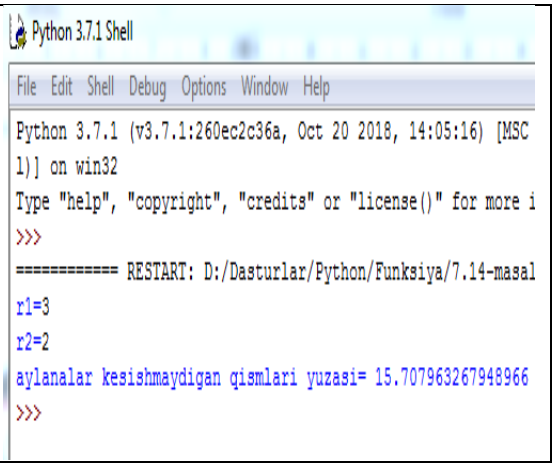
9.13-masala. Doiraning yuzini hisoblovchi funksiya hosil qiling. Bu funksiya yordamida 3 ta doira yuzini hisoblang. Doiraning yuzi $S = \pi R^2$ formula orqali hisoblanadi. $\pi = 3.1415$ ni o‘zgarmas deb qabul qiling.

```
import math;
def Doiraning_yuzi(r):
    S=math.pi*math.pow(r,2);
    return S;
a=int(input('r1='));
b=int(input('r2='));
c=int(input('r3='));
print('r sonning
darajasi=',Doiraning_yuzi(a));
print('r sonning
darajasi=',Doiraning_yuzi(b));
print('r sonning
darajasi=',Doiraning_yuzi(c));
```



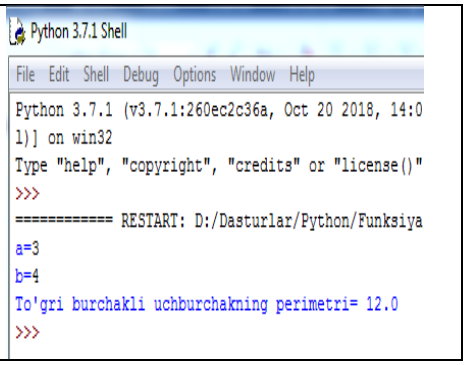
9.14-masala. Markazi bir nuqtada bo‘lgan, R1 va R2 radiusga ega 2 ta aylananing ustma - ust tushmaydigan (kesishmaydigan) qismining yuzasini topuvchi RingS nomli funksiya hosil qiling. Doiraning yuzini hisoblash formulasidan foydalaning. $S = \pi R^2$, $\pi = 3.1415$ ni o‘zgarmas deb qabul qiling.

```
import math;
def RingS(a,b):
    S=abs(math.pi*math.pow(a,2)-
math.pi*math.pow(b,2));
    return S;
a=int(input('r1='));
b=int(input('r2='));
print('aylanalar kesishmaydigan qismlari
yuzasi=',RingS(a,b));
```



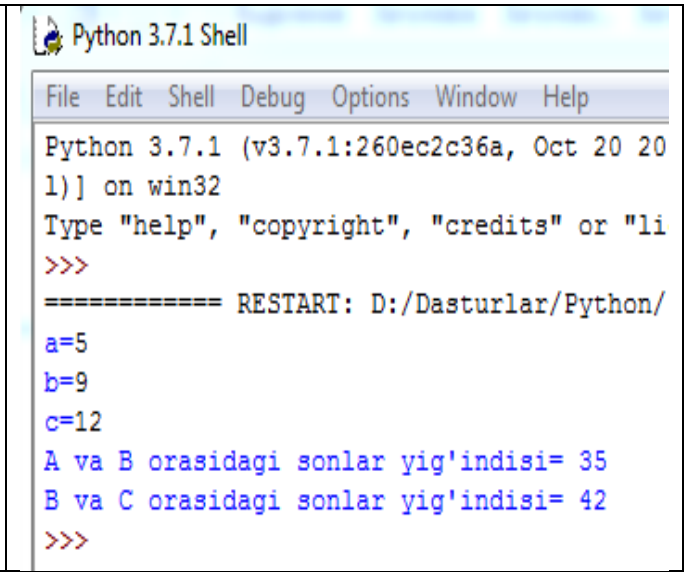
9.15-masala. To‘g‘ri burchakli uchburchakning katetlari a va b berilganda. uning perimetrini hisoblovchi TriangleP nomli funksiya hosil qiling.

```
import math;
def Triangle(a,b):
    p=a+b+math.sqrt(math.pow(a,2)+
+math.pow(b,2));
    return p;
a=int(input('a=')); b=int(input('b='));
print("To'gri burchakli uchburchakning
perimetri=",Triangle(a,b));
```



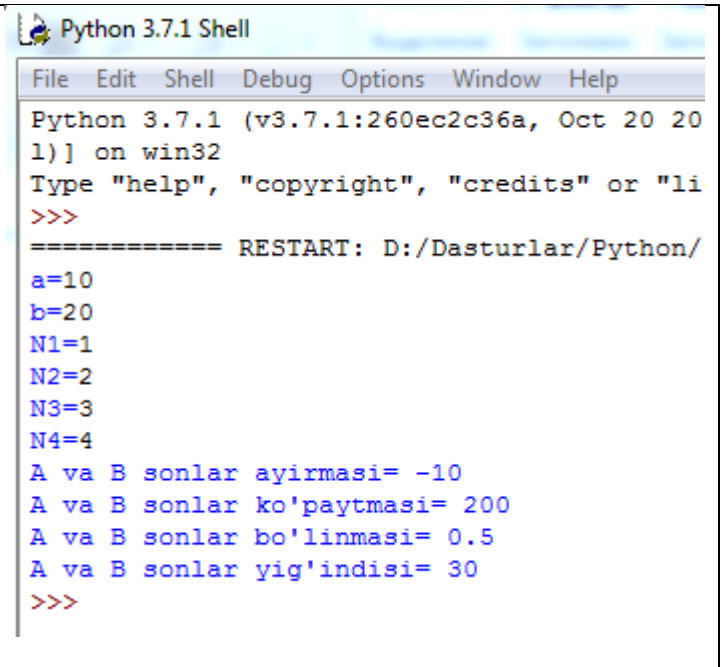
9.16-masala. A va B sonlari orasidagi sonlar yig‘indisini hisoblovchi SumRange(A,B) nomli funksiya hosil qiling. Agar A>B bo‘lsa. funksiya 0 qiymat qaytaradi. Bu funksiya orqali A dan B gacha va B dan C gacha bo‘lgan sonlar yig‘inddisini hisoblang, A, B, C butun sonlar.

```
def SumRange(x,y):
    S=0;
    while x<=y:
        S+=x;
        x+=1;
    return S;
a=int(input('a='));
b=int(input('b='));
c=int(input('c='));
print("A va B orasidagi sonlar
yig'indisi=",SumRange(a,b));
print("B va C orasidagi sonlar
yig'indisi=",SumRange(b,c));
```



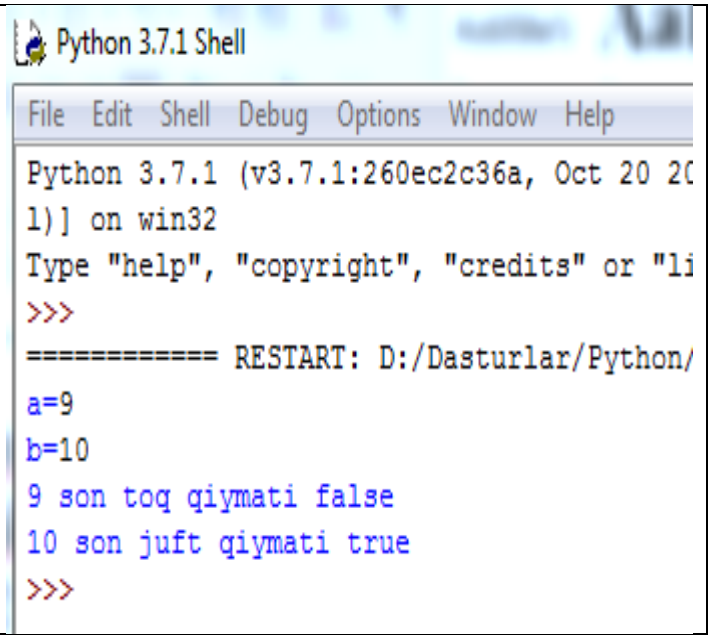
9.17-masala. Arifmetik amallarni bajaruvchi Calc(A, B, Op) funksiyasini hosil qiling. A va B haqiqiy sonlar. Op o‘zgaruchisi orqali bajariladigan arifmetik amal aniqlanadi. 1 - ayirish, 2 - ko‘paytirish, 3 - bo‘lish, boshqalari qo‘shish. Shu funksiya orqali A va B sonlari uchun N1, N2, N3, N4 amallari bajarilsin. (N1 - N4 butun sonlar)

```
def Calc(a,b,op):
    if op==1:       S=a-b;
    elif op==2:     S=a*b;
    elif op==3:     S=a/b;
    elif op==4:     S=a+b;
    return S;
a=int(input('a='));
b=int(input('b='));
n1=int(input('N1='));
n2=int(input('N2='));
n3=int(input('N3='));
n4=int(input('N4='));
print('A va B sonlar
ayirmasi=',Calc(a,b,n1));
print("A va B sonlar
ko'paytmasi=",Calc(a,b,n2));
print("A va B sonlar
bo'linmasi=",Calc(a,b,n3));
print("A va B sonlar
yig'indisi=",Calc(a,b,n4));
```



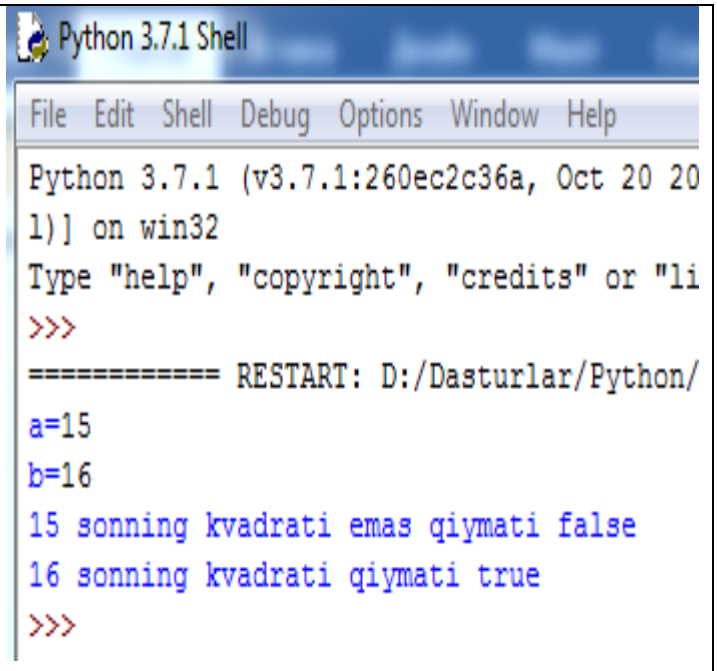
9.18-masala. Butun sonning juft yoki toqligini aniqlovchi Even(K) funksiyasini hosil qiling. Funksiya K juft son bo‘lsa - true, aks holda false qiymat qaytarsin. Bu funksiya orqali 2 ta sonning juft yoki toqligi aniqlansin.

```
def Even(k):
    return k%2==0;
a=int(input('a='));
b=int(input('b='));
if Even(a):
    print(a,"son juft qiymati true");
else:
    print(a,"son toq qiymati false");
if Even(b):
    print(b,"son juft qiymati true");
else:
    print(b,"son toq qiymati false");
```



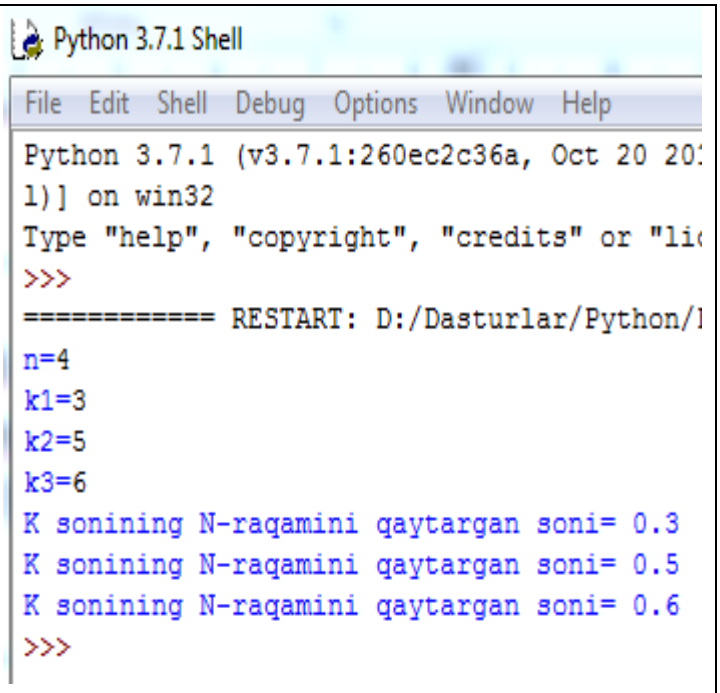
9.19-masala. IsSquare(K) mantiqiy funksiyasini hosil qiling, (K>0). Agar K biror butun sonning kvadrati bo‘lsa - true, aks holda false qiymat qaytarilsin. Shu funksiya orqali 2 ta sonni tekshiring.

```
import math;
def IsSquare(k):
    ildiz=math.sqrt(k);
    if ildiz*ildiz==k:
        return True;
    else:
        return False;
a=int(input('a='));
b=int(input('b='));
if IsSquare(a):
    print(a,'sonning kvadrati qiymati
true');
else:
    print(a,'sonning kvadrati emas
qiymati false');
if IsSquare(b):
    print(b,'sonning kvadrati qiymati
true');
else:
    print(b,'sonning kvadrati emas
qiymati false');
```



9.20-masala. Butun qiymat qaytaruvchi DigitN(K,N) funksiyasini hosil qiling, (K>0). Funksiya K sonining N - raqamini qaytarsin. Agar K soni raqamlari N dan kichk bo‘lsa. minus bir qaytarilsin. Shu funksiya orqali K1, K2. K3 sonlarining N - raqami aniqlansin.

```
def DigitCount(K):
    soni=0;
    while K>0:
        soni+=1;
        K/=10;
    return soni;
def DigitN(K,N):
    soni=DigitCount(K);
    if soni<N:
        return -1;
    elif soni==N:
        return K%10;
    else:
        return K/10;
n=int(input('n='));
k1=int(input('k1='));
k2=int(input('k2='));
k3=int(input('k3='));
print("K sonining N-raqamini
qaytargan soni=",DigitN(k1,n));
print("K sonining N-raqamini
qaytargan soni=",DigitN(k2,n));
print("K sonining N-raqamini
qaytargan soni=",DigitN(k3,n));
```



6.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLARI

Topshiriq: 1) Quyidagi masalalarning PYTHON tilidagi dasturini funksiyadan foydalanib tuzing:

9.1-masala. Ihtiyoriy sonning 3 - darajasini hisoblovchi PowerA3 nomli funksiya hosil qiling. PowerA3 funksiyasi orqali A, B, C haqiqiy sonlarining va D, E butun sonlarining 3 - darajasini hisoblovchi dasturini tuzing.
9.2-masala. Ihtiyoriy sonning 2, 3, 4 - darajasini hisoblovchi PowerA234 nomli funksiya hosil qiling. PowerA234 funksiyasi orqali A, B, C haqiqiy sonlarining 2, 3, 4 - darajasini hisoblovchi dasturini tuzing. Funksiya prototipi quyidagicha bo‘lishi mumkin: PowerA234(float . float *, float *, float *);
9.3-masala. 2 ta sonning o‘rta arifmetigi va goemetrigini hisoblovchi MEAN nomli funksiya hosil qiling. MEAN funksiyasi orqali A, B, C, D sonlaridan (A. B), (A. C), (A. D) juftliklarining o'rta arifmetigi va geometrigini hisoblovchi dasturni tuzing.
Funksiya prototipi quyidagicha bo‘lishi mumkin: MEAN(float, float . float *, float *);
9.4-masala. Teng tomonli 3 burchakning yuzasi va perimetrini hisoblovchi Triangle nomli funksiya hosil qiling. Triangle funksiyasi orqali 3 ta teng tomonli uchburchakning perimetri va yuzini hisoblovchi dasturni tuzing. Triangle(float, float *, float *);
9.5-masala. To‘g‘ri to‘rtburchakning yuzini va perimetrini uning qarama - qarshi uchlari koordinatasi orqali hisoblovchi RectPS nomli funksiya hosil qiling. (x1,y1,x2,y2) to‘g‘ri to‘rtburchakning qarama - qarshi uchlari. RectPS funksiyasi orqali 2 ta to‘rtburchak yuzi va perimetrini hisoblang. To‘rtburchak tomonlari koordinatalar o‘qiga parallel. Funksiya prototipi quyidagicha bo‘lishi mumkin: RectPS(int, int, int *, int *);
9.6-masala. Natural sonning raqamlari soni va raqamlari yig‘indisini hisoblovchi DigitCountSum nomli funksiya hosil qiling. Bu funksiya orqali a, b, c sonlarining raqamlari soni va yig‘indisini hisoblovchi dasturni tuzing. DigitCountSum **(**int, int ***,** int ***);**
9.7-masala. Butun musbat sonining raqamlarini teskari tartibda chiqaruvchi InvertDigit nomli funksiya hosil qiling. Bu funksiya orqali a, b, c sonlarining raqamlarini teskari tartibda chiqaruvchi dasturni tuzing. Funksiya prototipi quyidagicha bo‘lishi mumkin: InvertDigit **(int**);
9.8-masala. Kiritilgan K butun musbat sonining o‘ng tarafiga (oxiriga) R raqamini (1<=R<=9) qo‘shuvchi AddRightDigit nomli funksiya hosil qiling. Funksiya prototipi quyidagicha bo‘lishi mumkin: AddRightDigit **(int** son, **int** raqam);
9.9-masala. Kiritilgan K butun musbat sonining chap tarafiga (boshiga) R raqamini (1<=R<=9) qo‘shuvchi AddLeftDigit nomli funksiya hosil qiling. Funksiya prototipi quyidagicha bo‘lishi mumkin: AddLeftDigit **(int** *son, **int** raqam);
9.10-masala. Ikkita sonning qiymatini almashtiruvchi Swap nomli funksiya hosil qiling. Swap funksiyasi orqali A, B, C, D sonlaridan (A, B), (D, C) juftliklarining qiymatlarini almashtiruvchi dasturini tuzing. Funksiya prototipi quyidagicha bo‘lishi mumkin: Swap **(int *, int *);**

9.11-masala. X va Y sonlaridan kichigini X ga va kattasini Y ga yozuvchi Minmax(X, Y) funksiyasini hosil qiling. Minmax funksiyagini 4 marta chaqarish orqali a. b, c, d butun sonlaridan kattasini va kichigini aniqlovchi dasturni tuzing.

9.12-masala. A. B. C sonlarini o‘sish tartibida joylashtiruvchi Sortlnc3(A, B, C) funksiyasini hosil qiling. Ya’ni A. B. C sonlari qiymatlarini shunday almashtiringki, natijada A ning qiymati eng kichik va C ning qiymati eng katta bo‘lsin. Bu funksiya orqali (A1, B1, C1) va (A2, B2, C2) sonlarini tartiblang.

9.13-masala. A, B, C sonlarini kamayish tartibida joylashtiruvchi SortDec3(A, B, C) funksiyasini hosil qiling. Ya’ni A. B, C sonlari qiymatlarini shunday almashtiringki, natijada A ning qiymati eng katta va C ning qiymati eng kichik bo‘lsin. Bu funksiya orqali (A1, B1, C1) va (A2, B2, C2) sonlarini tartiblang.

9.14-masala. O‘ngga siklik siljishni amalga oshiruvchi ShiftRight3(A. B. C) funksiyasini hosil qiling. Ya’ni A ning qiymati B ga, B ning qiymati C ga, C ning qiymati A ga o‘tib qolsin. Bu funksiya orqali (A1, B1, C1) va (A2. B2, C2) sonlarini siljiting.

9.15-masala. Chapga siklik siljishni amalga oshiruvchi ShiftLeft3(A, B, C) funksiyasini hosil qiling. Ya’ni C ning qiymati B ga, B ning qiymati A ga, A ning qiymati C ga o‘tib qolsin. Bu funksiya orqali (A1, B1, C1) va (A2, B2, C2) sonlarini siljiting.

9.16-masala. Haqiqiy sonning ishorasini aniqlovchi ishora nomli funksiya hosil qiling. Funksiya argumenti noldan kichik bo'lsa -1; noldan katta bo‘lsa 1; nolga teng bo‘lsa 0 qiymat qaytarsin. Haqiqiy a va b sonlari uchun ishora(a)+ishora(b) ifodasi hisoblansin.

9.17-masala. Kvadrat tenglamaning ildizlar sonini aniqlovchi funksiya hosil qiling. $a * x^2 + b * x + c = 0$ ko‘rinishidagi tenglama kvadrat tenglama deyiladi. (a noldan farqli son) dasturini tuzing.

9.18-masala. Doiraning yuzini hisoblovchi funksiya hosil qiling. Bu funksiya yordamida 3 ta doira yuzini hisoblang. Doiraning yuzi $S = \pi * R^2$ orqali hisoblanadi, π=3.1415 ni o‘zgarmas deb qabul qiling.

9.19-masala. Markazi bir nuqtada bo‘lgan, R1 va R2 radiusga ega 2 ta aylananing ustma - ust tushmaydigan (kesishmaydigan) qismining yuzasini topuvchi RingS nomli funksiya hosil qiling. Doiraning yuzini hisoblash formulasidan foydalaning. $S = \pi * R^2$, π = 3.1415 ni o‘zgarmas deb qabul qiling.

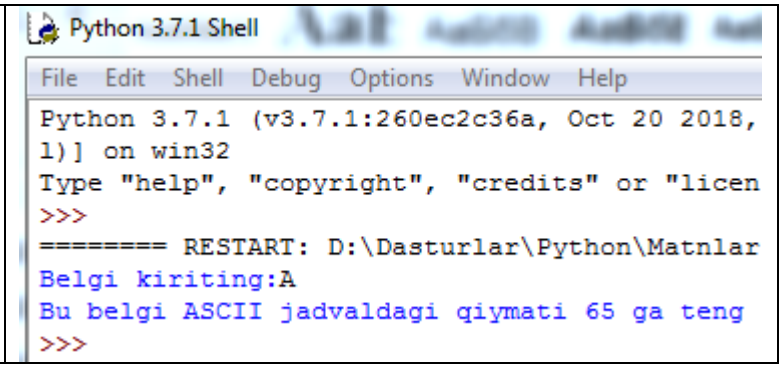
9.20-masala. To‘g‘ri burchakli uchburchakning katetlari A va B berilganda, uning perimetrini hisoblovchi TriangleP nomli funksiya hosil qiling.

VII. BOB. PYTHON DA MATNLAR BILAN ISHLASH

7.1 SIMVOLLARNI TAQQOSLASH

Har bir simvol ASC II – standart kodiga ega. ASC II – ushbu qisqartma so‘z American Standard Code for Information Interchange (Ma’lumotlar almashinish bo‘yicha Amerika standarti) inglizcha so‘zining birinchi harflaridan olingan u simvollarni kodlashtirishning universal sxemasi bo‘lib, shu sohadagi standartni (andozani) belgilaydi. Ixtiyoriy simvolning ACS II – kodini aniqlash uchun simvolni ord() PYTHON – funktsiyaning argumenti sifatida beradi. Satr harflari “a” dan “z” gacha, 97-122 qiymatlari orasida, bosh harflar “A”-“Z” esa 65-90 gacha bo‘lgan qiymatlar orasida joylashgan. Har bir oraliqdagi birinchi qiymatlarni taqqoslab ko‘rish mumkinki, satr harf “a” (97), bosh harf “A” (65) ga qaraganda kattaroq kodga ega. Shu sababli, uni strcmp() funktsiyani argument ko‘rinishiga keltirsak, strcmp() funktsiya 1 ni qaytaradi, chunki birinchi argumentning qiymati ikkinchisinikidan katta. Boshqa tomondan, argument “A” (65) ni birinchi argument sifatida “a” (97) ni esa ikkinchi argument sifatida ko‘rsatsak u holda strcmp() funktsiya -1 ni qaytaradi, chunki birinchi argumentning qiymati ikkinchisinikidan kichik. Satrlarni taqqoslashda har bir simvol ketma – ketligidagi o‘rni bo‘yicha taqqoslanadi, bunda aynan bir simvolni saqlovchi satrlar, turli tartibda joylashgan bo‘lsa, ularning xatoliklari teng bo‘lib qolmasligini kafolatlaydi. Masalan, “ABC” va “BAC” satrlarni taqqoslashda, birinchi satrning birinchi simvoli “A” (65), ikkinchi satrning birinchisimvoli “B” (66) ga qaraganda kichik bo‘ladi, shu sababli, strcmp() funktsiya -1 ni qaytaradi, chunki birinchi argument ikkinchisidan kichik.

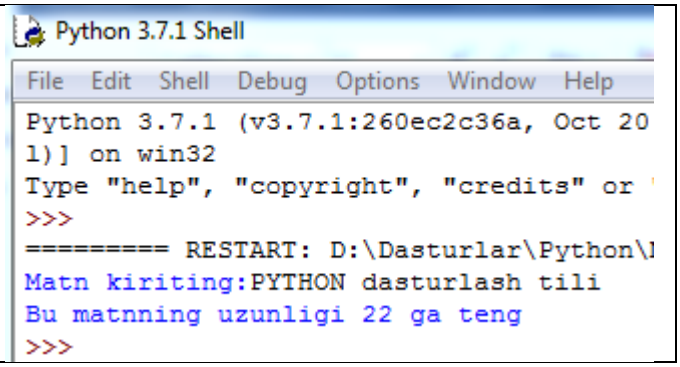
```
belgi=input('Belgi kiriting:');
s=ord(belgi);
print("Bu belgi ASCII jadvaldagi qiymati",s,'ga teng');
```



Eslatib o‘tish lozimki, ASC II – kodning umumiy hajmi uchun simvollarning joylashishi tartibining ahamiyati yo‘q: turli tartibli so‘zlar (yoki harflar) bilan yozilgan ikkita satr kod hajmi bo‘yicha ekvivalent bo‘ladi, ammo, bir – biriga mos tushmasligi mumkin.

Satr uzunligini esa strlen() PYTHON – funktsiyaning argumenti sifatida ko‘rsatish orqali aniqlash mumkin.

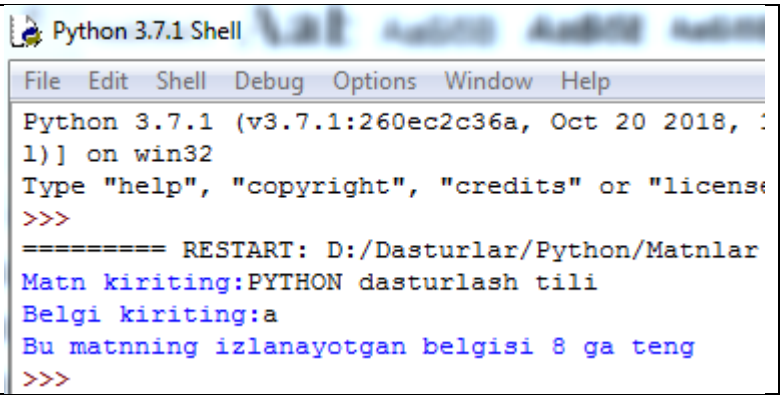
```
belgi=input("Matn kiriting:");
s=len(belgi);
print("Bu matnning uzunligi",s,'ga
teng');
```



7.2 MATNNI IZLASH

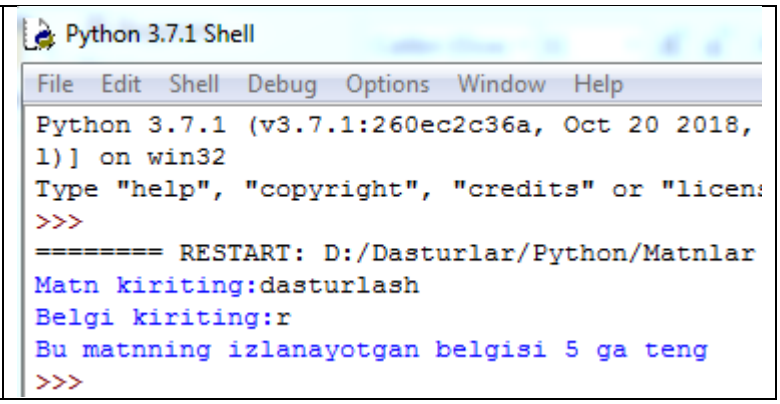
PYTHON tilida bir nechta tuzilgan funktsiyalar mavjud bo‘lib, ular satrlar ichidagi ma’lum qismiy satrni yoki alohida simvollarni izlashga imkon beradi. Ushbu sodda funktsiya find() ikkita argumentni qabul qiladi, ular o‘z navbatiga izlanishi lozim bo‘lgan satrni va izlanayotgan qism satrni ko‘rsatadi. Izlash satr boshidan to mos tushgan simvolni topguncha ketma – ket amalga oshiriladi. Agar mos tushush aniqlansa, izlash to‘xtaydi va find() funktsiya butun sonni qaytaradi, u satrdagi simvol indeksidan iborat bo‘lib, birinchi mos tushushni aniqlaydi. Agar mos tushush aniqlanmasa, find() funktsiya false qiymatni qaytaradi.

```
matn=input('Matn kiriting:');
belgi=input('Belgi kiriting:');
s=matn.find(belgi);
print('Bu matnning izlanayotgan
belgisi',s,'ga teng');
```

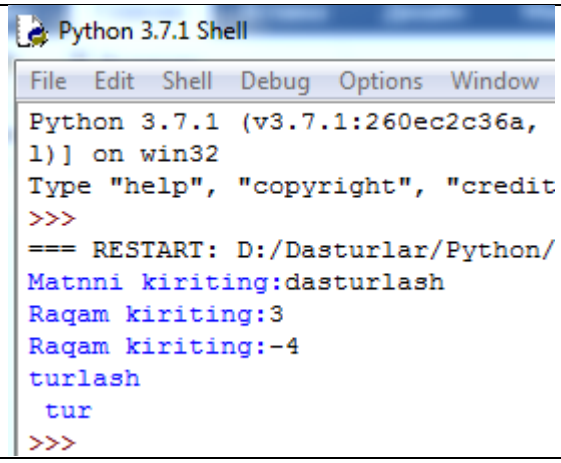


rfind() funktsiya ham xuddi shu tariqa ishlaydi. Ammo, satrning oxiridan teskari tartibda izlashda simvol indeksini qaytaradi.

```
matn=input('Matn kiriting:');
belgi=input('Belgi kiriting:');
s=matn.rfind(belgi);
print('Bu matnning izlanayotgan
belgisi',s,'ga teng');
```



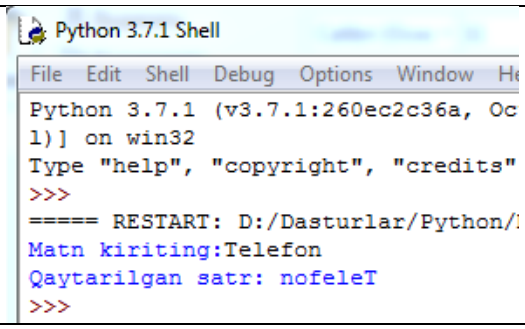
```
matn=str(input('Matnni kiriting:'));
raqam1=int(input('Raqam kiriting:'));
raqam2=int(input('Raqam kiriting:'));
s=matn[raqam1:len(matn)];
s1=matn[raqam1:raqam2];
print(s,'\n',s1);
```



7.3 SATRLARNI FORMATLASH

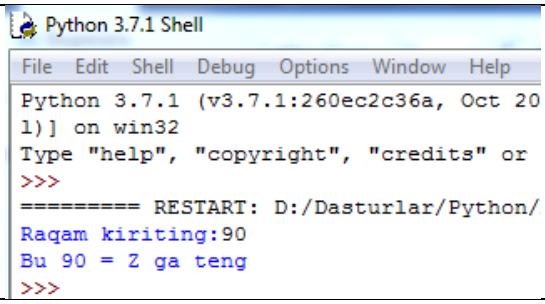
PYTHON tilida simvollarni boshqarish orqali satrlarni qulay formatlashga mo‘ljallangan bir nechta tuzilgan funktsiyalar mavjud. Ushbu sodda strrev() funktsiya bitta satrli argumentni qabul qiladi va ushbu satrni simvollarning teskari tartibda joylashuvini qaytaradi, qisqa qilib aytganda “orqadan oldinga qarab o‘qiydi”.

```
matn=str(input("Matn kiriting:"));
s=matn[::-1];
print("Qaytarilgan satr:",s);
```



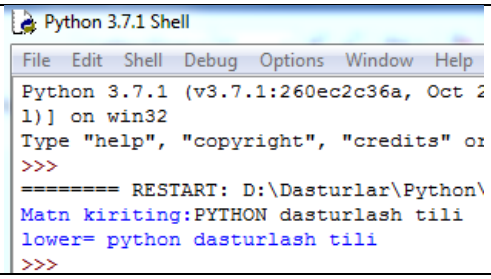
- chr() – simvolni uning kodi bo‘yicha qaytaradi.

```
son=int(input("Raqam kiriting:"));
s=chr(son);
print('Bu',son,'=',s,'ga teng');
```



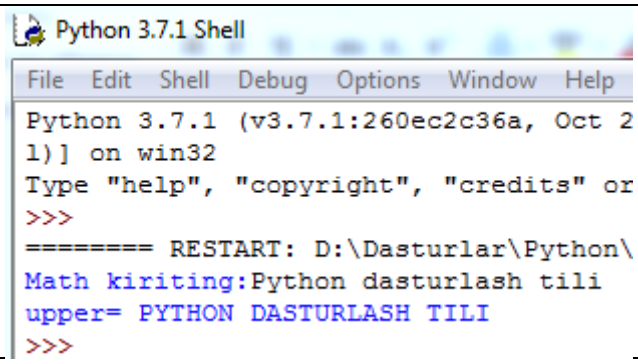
- lower() – satrni quyi registrga aylantiradi. Katta harflarni kichik harflarga almashtiradi.

```
matn=input('Matn kiriting:');
s=matn.lower();
print('lower=',s);
```



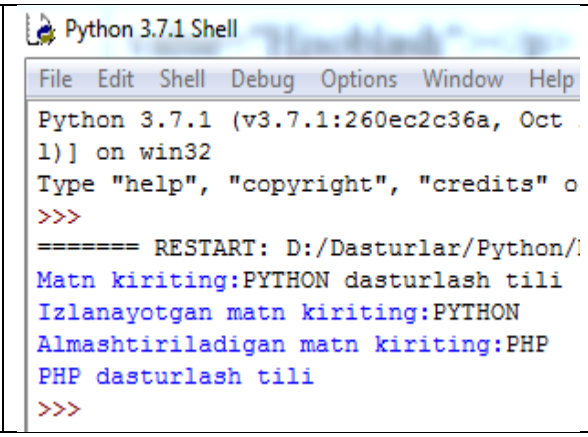
- upper() – satrni yuqori registrga aylantiradi. Kichik harflarni katta harflarga almashtiradi.

```
matn=input('Math kiriting:');
s=matn.upper();
print('upper=',s);
```



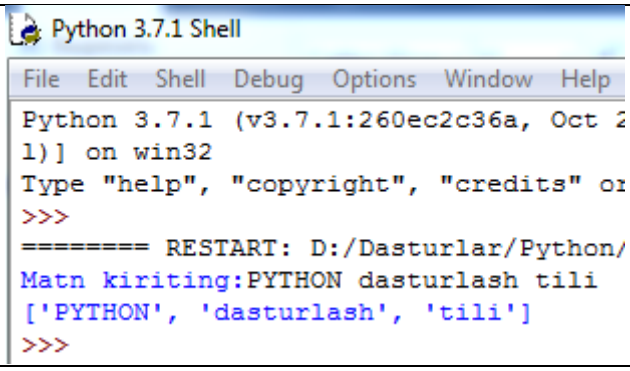
- replace() – izlanayotgan satrni almashtirilishi lozim bo‘lgan satrga almashtiradi.

```
matn1=str(input("Matn kiriting:"));
matn2=str(input("Izlanayotgan
matn kiriting:"));
matn3=str(input("Almashtiriladigan
matn kiriting:"));
s=matn1.replace(matn2,matn3);
print(s);
```



- split() – satrni massivga aylantiradi.

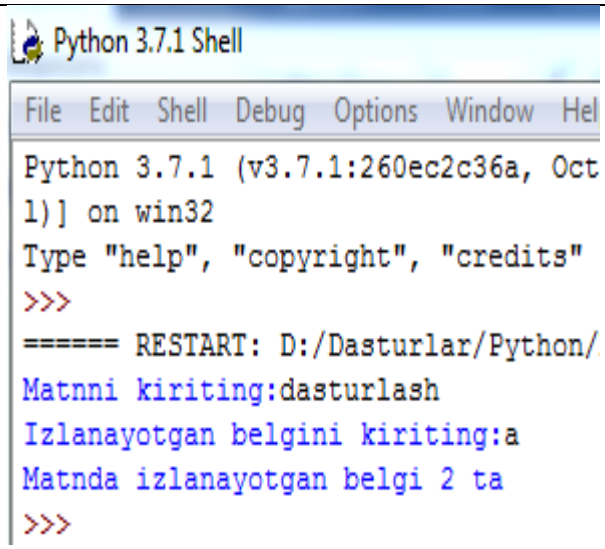
```
matn=str(input('Matn kiriting:'));
massiv=matn.split(" ");
print(massiv);
```



7.4 SATRLAR BILAN ISHLASH FUNKTSIYASI VA UNING TADBIQI

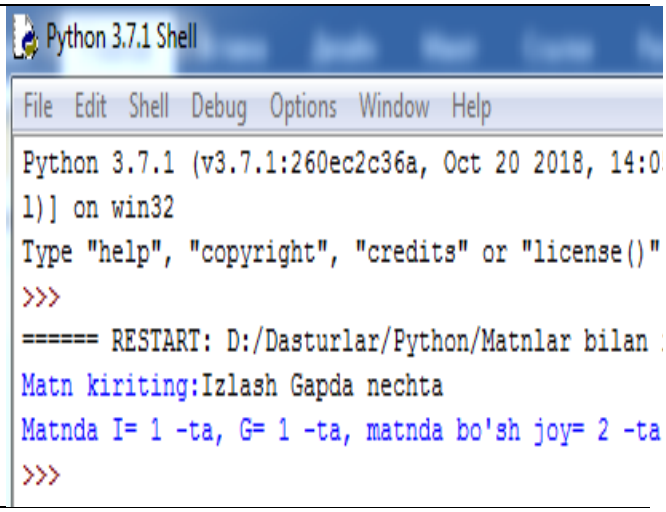
10.1-masala. N ta so‘zdan tashkil topgan matnda berilgan so‘z necha marta uchrashini aniqlang.

```
matn=str(input('Matnni kiriting:'));
belgi=str(input('Izlanayotgan belgini
kiriting:'));
m=len(matn);
n=len(belgi);
b=0;
for i in range((m-n)+1):
    if matn[i:i+1]==belgi:
        b=b+1;
if b==0:
    print("Matnda izlanayotgan belgi
yo'q");
else:    print("Matnda izlanayotgan
belgi",b,"ta");
```



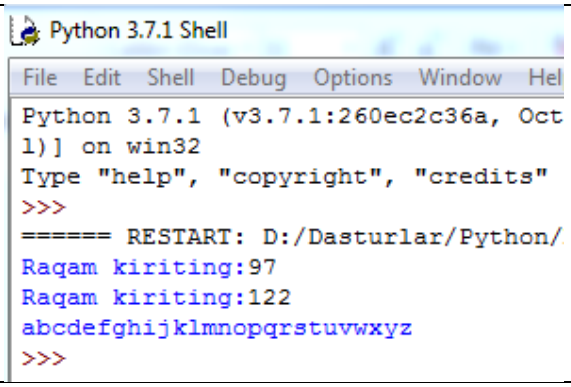
10.2-masala. Berilgan matndagi G va I harflar hamda bo‘sh joylar sonini aniqlang.

```
matn=str(input("Matn kiriting:"));
m=0;n=0;k=0;
for i in range(len(matn)):
    S=matn[i:i+1];
    if S=='I':
        n=n+1;
    if S=='G':
        m=m+1;
    if S==' ':
        k=k+1;
print("Matnda I=",n,'-ta, G=',m,"-ta,
matnda bo'sh joy=",k,'-ta');
```



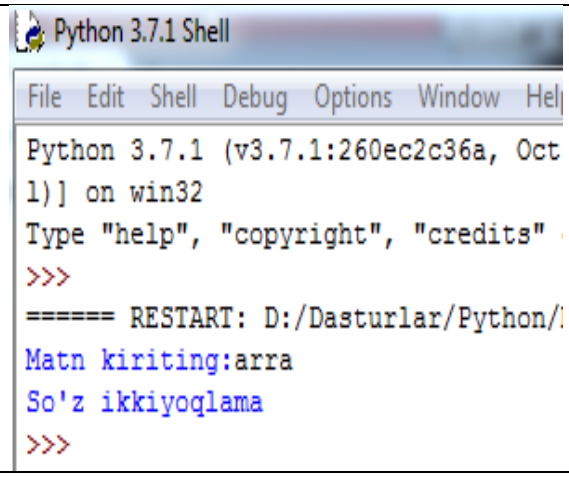
10.3-masala. ASC II jadvalidan kichik lotin harflarni chiqaring.

```
n=int(input('Raqam kiriting:'));
m=int(input('Raqam kiriting:'));
for i in range(n,m+1):
    print(chr(i),end='');
```



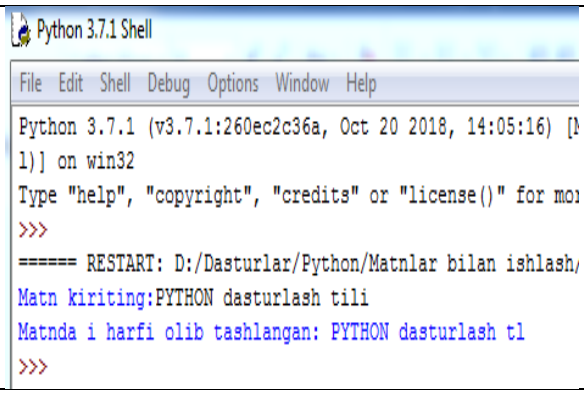
10.4-masala. Berilgan so‘z ikkiyoqlama bo‘lishini aniqlang.

```
import math;
matn=str(input("Matn kiriting:"));
s=len(matn);t=0;
for i in range(1,math.ceil(s/2)):
    if matn[:i]==matn[-i:]:
        t=t+1;
        print("So'z ikkiyoqlama");
    elif t==s%2:
        print("So'z ikkiyoqlama emas");
```



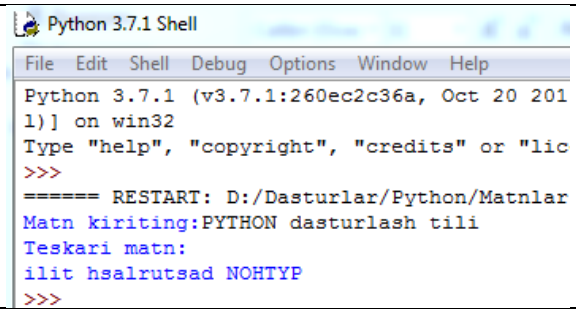
10.5-masala. Berilgan matndagi hamma I harflarni olib tashlang.

```
matn=str(input("Matn kiriting:"));
olish='i';
s=matn.replace(olish,'');
print("Matnda i harfi olib tashlangan:",s);
```



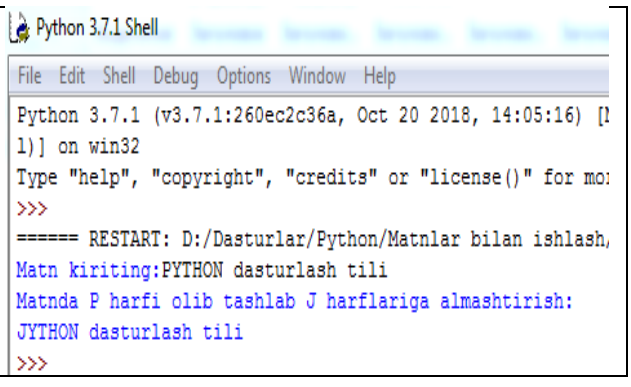
10.6-masala. Berilgan matnni teskarisi tartibda yozing.

```
matn=str(input('Matn kiriting:'));
s=matn[::-1];
print("Teskari matn:");
print(s);
```



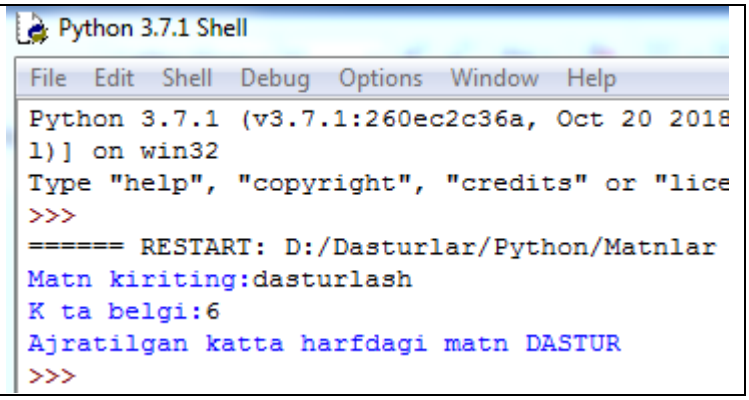
10.7-masala. Berilgan matndagi P harflarni J harflarga almashtiring.

```
matn=str(input("Matn kiriting:"));
s=matn.replace("P","J");
print("Matnda P harfi olib tashlab J
harflariga almashtirish:");
print(s);
```



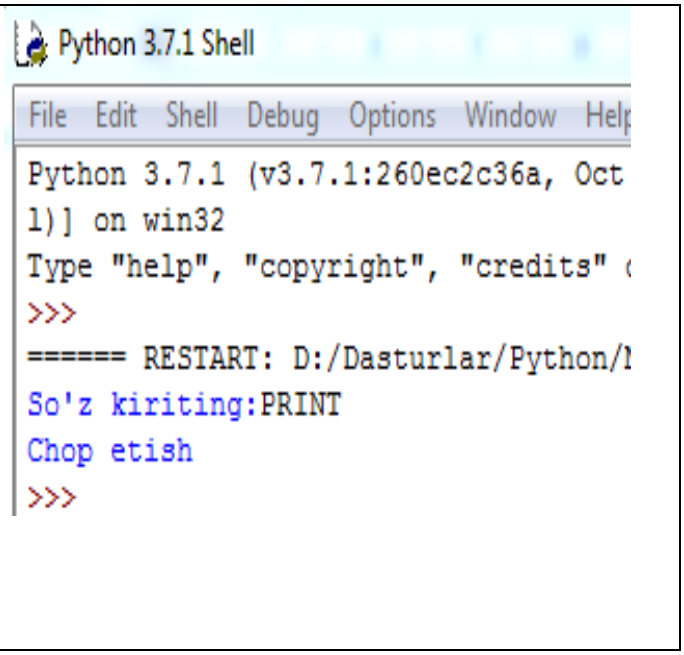
10.8-masala. Matnda uzunligi K ta belgidan katta bo‘lgan so‘zlarni ajratilib yangi matnga yozish dasturini tuzing**.**

```
matn=str(input('Matn kiriting:'));
k=int(input('K ta belgi:'));
s=matn[0:k];
print('Ajratilgan katta harfdagi
matn',s.upper());
```



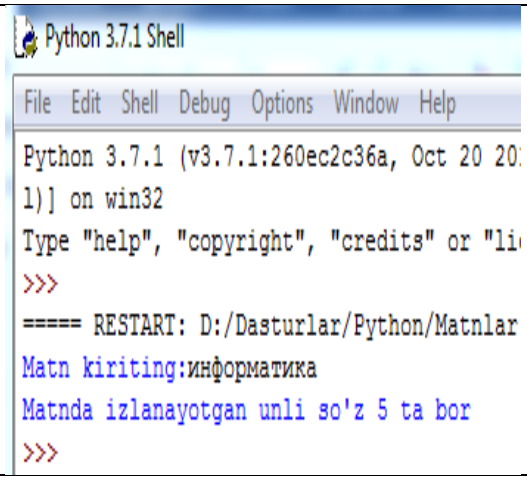
10.9-masala. Inglizcha - o‘zbekcha lug‘atni tuzing. Bunda inglizcha so‘z kiritilganda uning tarjimasi natija sifatida olinishini ta’minlang.

```
matn=str(input("So'z kiriting:"));
if matn=='REM':
    print("Izoh");
if matn=='IF':
    print('Agar');
if matn=='FOR':
    print('Uchun');
if matn=='INPUT':
    print('Kiritish');
if matn=='STOP':
    print("To'xta");
if matn=='PRINT':
    print("Chop etish");
if matn=='RUN':
    print("Bajar");
```



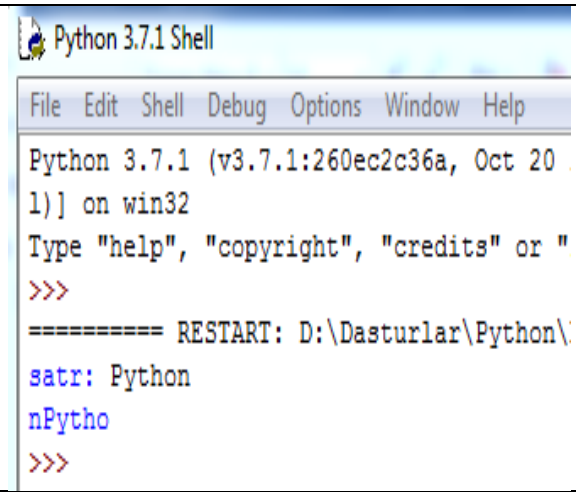
10.10-masala. Matndagi so‘zlarda nechta unli harflar borligini aniqlovchi dastur tuzing.

```
matn=str(input('Matn kiriting:'));
unli='аоиуеэяўё';
s=0;
for i in range(len(matn)):
    for j in range(len(unli)):
        if matn[i:i+1]==unli[j:j+1]:
            s=s+1;
print("Matnda izlanayotgan unli
so'z",s,'ta bor');
```



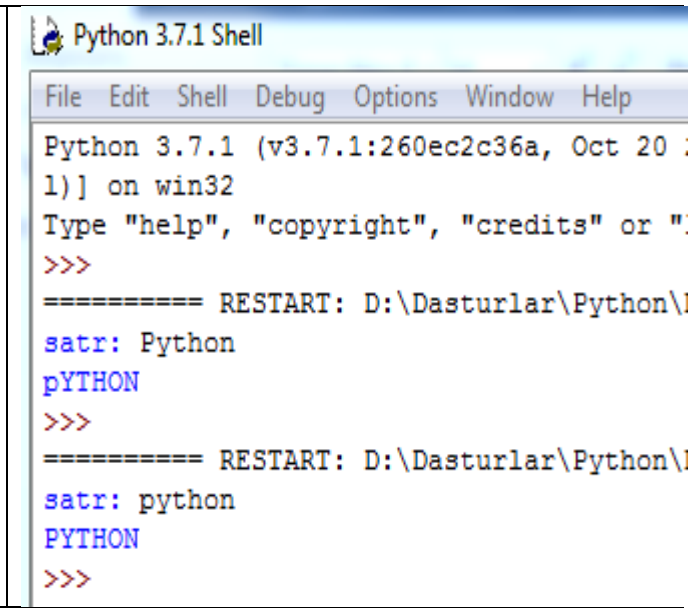
10.11-masala. Satr berilgan. Satrning oxirgi belgisini birinchi belgi qiluvchi dastur tuzing.

```
satr = input('satr: ')
satr = list(satr)
a = []
a.append(satr.pop(-1))
a.extend(satr)
satr = ''
for i in a:
    satr += i
print(satr)
```



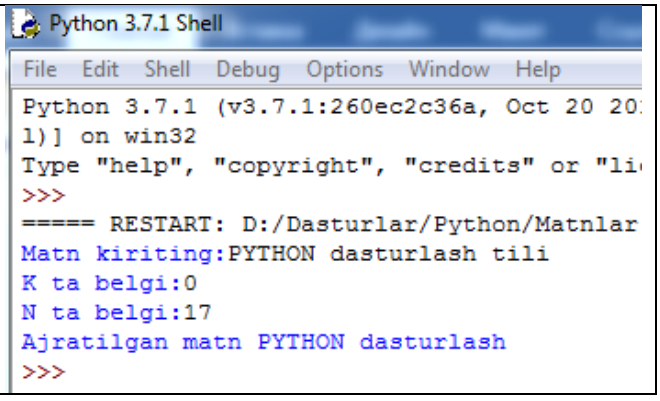
10.12-masala. Satr berilgan. Ushbu satrdagi kichik harflarni katta harflarga o‘giruvchi dastur tuzing.

```
satr = input('satr: ')
l = list(satr)
for j in range(len(l)):
    if 65 <= ord(l[j]) and ord(l[j]) <= 90:
        l[j] = chr(ord(l[j]) + 32)
    elif 97 <= ord(l[j]) and ord(l[j]) <=
122:
        l[j] = chr(ord(l[j])-32)
satr =''
for i in l:
    satr += i
print(satr)
```



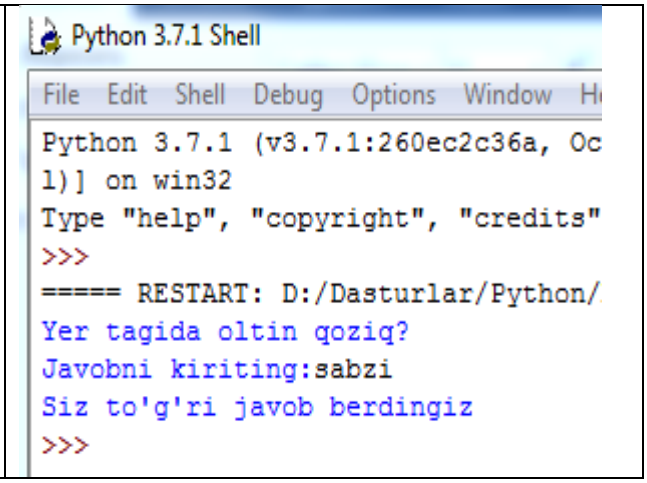
10.13-masala. Berilgan matnning orasidagi K-simvoldan N-simvolgacha bo‘lgan belgilarni ajrating**.**

```
matn=str(input('Matn kiriting:'));
BelgiK=int(input('K ta belgi:'));
BelgiN=int(input('N ta belgi:'));
s=matn[BelgiK:BelgiN-BelgiK];
print("Ajratilgan matn",s);
```



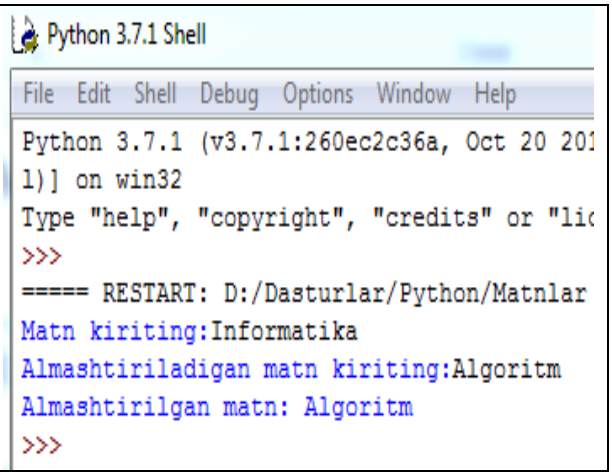
10.14-masala. Topishmoq topish va uning javobini tahlil qilish dasturini tuzing**.**

```
print("Yer tagida oltin qoziq?");
matn=str(input("Javobni kiriting:"));
if matn=='sabzi':
    print("Siz to'g'ri javob berdingiz");
else:
    print("Siz noto'g'ri javob berdingiz");
```



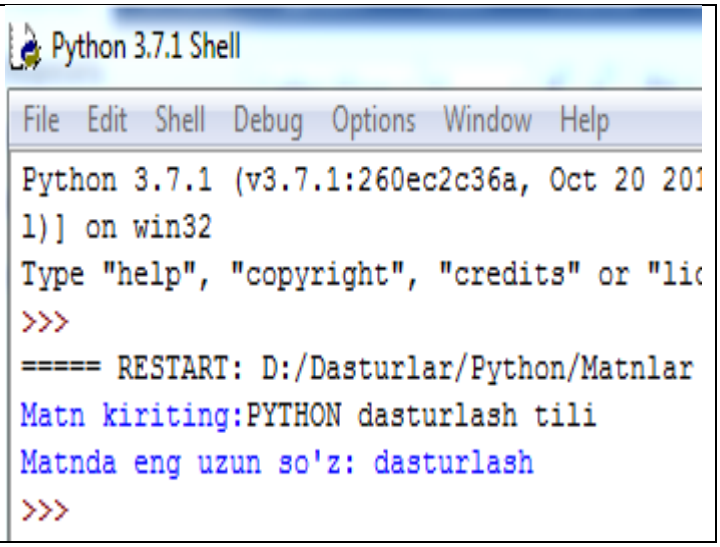
10.15-masala. Matndagi INFORMATIKA so‘zini ALGORITM so‘zi bilan almashtirish dasturini tuzing.

```
matn1=str(input("Matn kiriting:"));
matn2=str(input("Almashtiriladigan matn
kiriting:"));
s=matn1.replace(matn1,matn2);
print("Almashtirilgan matn:",s);
```



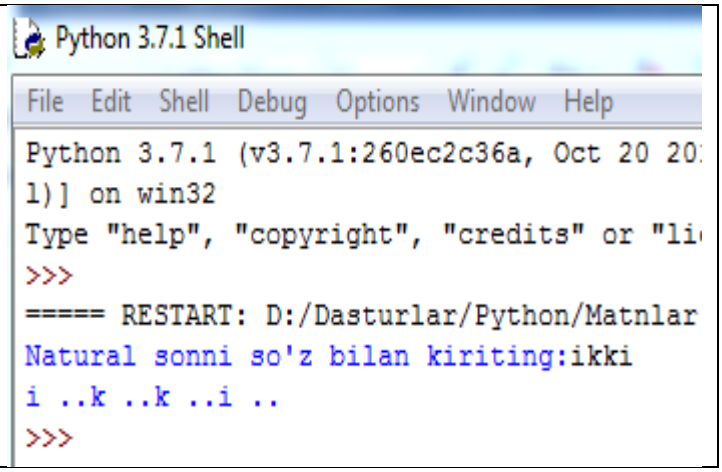
10.16-masala. Berilgan N ta so‘zlardan eng uzunini aniqlang.

```
matn=str(input("Matn kiriting:"));
s=matn.split(' ');
maks=len(s[0]);
for i in range(len(s)):
    if maks<len(s[i]):
        k=s[i];
print("Matnda eng uzun so'z:",k);
```



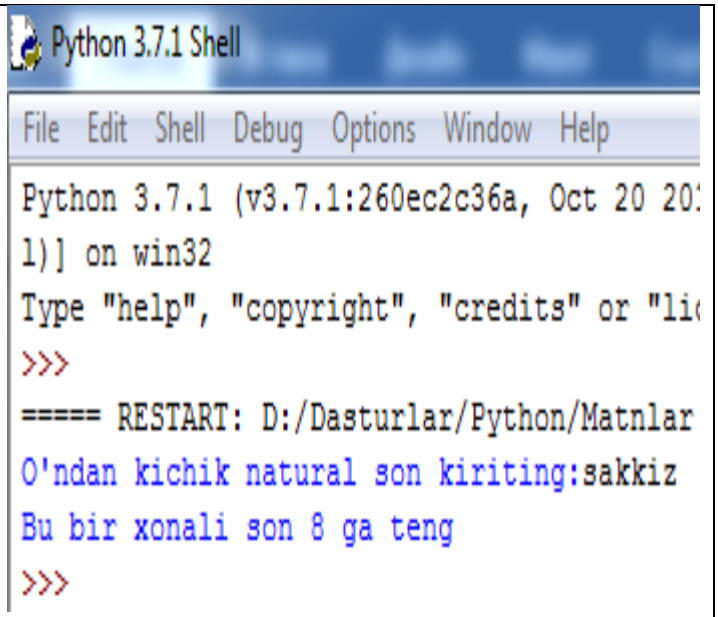
10.17-masala. Berilgan natural sonning xona birliklarini ajratib yozing.

```
matn=str(input("Natural sonni so'z
bilan kiriting:"));
s1=len(matn);
for i in range(s1):
    sn=matn[i:i+1];
    print(sn,'..',end='');
```



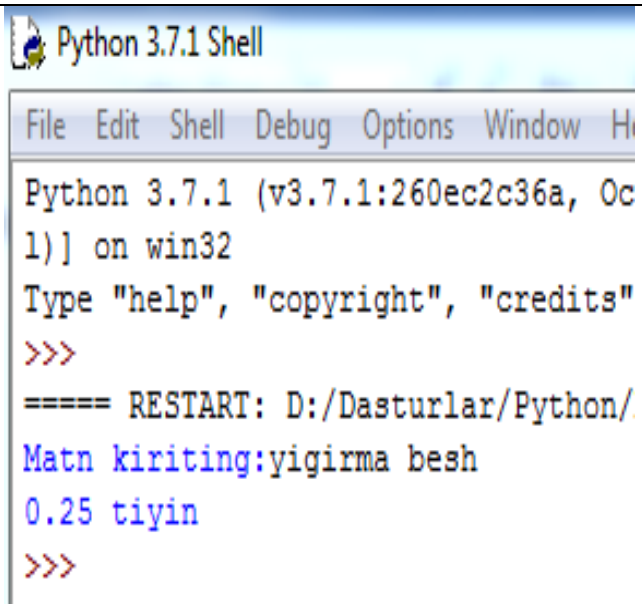
10.18-masala. O‘ndan kichik bo‘lgan so‘zlarni ularga mos natural sonda chop eting.

```
matn=str(input("O'ndan kichik
natural son kiriting:"));
n=len(matn);
if matn[0:n]=='bir':
    t=1;
elif matn[0:n]=='ikki':
    t=2;
elif matn[0:n]=='uch':
    t=3;
elif matn[0:n]=="to'rt":
    t=4;
elif matn[0:n]=="besh":
    t=5;
elif matn[0:n]=="olti":
    t=6;
elif matn[0:n]=="yetti":
    t=7;
elif matn[0:n]=="sakkiz":
    t=8;
elif matn[0:n]=="to'qqiz":
    t=9;
else:
    print("Bu bir xonali son emas yoki
natural son emas");
print("Bu bir xonali son",t,'ga teng');
```



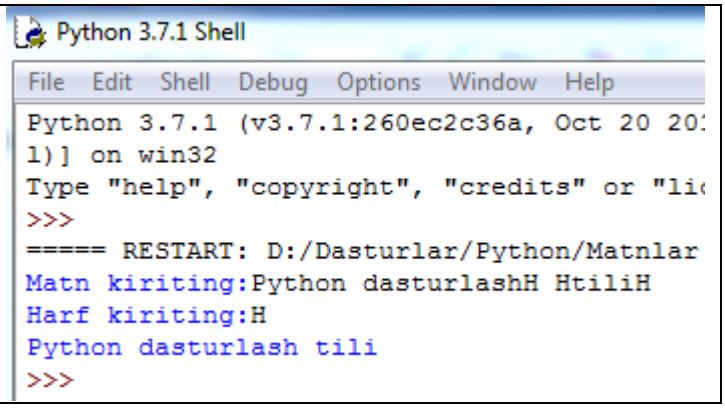
10.19-masala. Tiyinlarda berilgan pulni so‘m bilan ifodalang. Bunda tiyinlar ikki xonali sonlar bilan ifodalanadi.

```
m=str(input("Matn kiriting:"));
n=len(m);
for i in range(1,n):
    m1=m[:i];
    if m1=="o'n":       un=10;
    elif m1=="yigirma":
        un=20;
    elif m1=="o'ttiz":
        un=30;
    elif m1=="qirq":
        un=40;
    elif m1=="ellik":
        un=50;
    elif m1=="oltmish":
        un=60;
    elif m1=="yetmish":
        un=70;
    elif m1=="sakson":
        un=80;
    elif m1=="to'qson":
        un=90;
    m2=m[i:];
    if m2=='bir':
        bir=1;
    if m2=='ikki':
        bir=2;
    if m2=='uch':
        bir=3;
    if m2=="to'rt":
        bir=4;
    if m2=='besh':
        bir=5;
    if m2=='olti':
        bir=6;
    if m2=='yetti':
        bir=7;
    if m2=='sakkiz':
        bir=8;
    if m2=="to'qqiz":
        bir=9;
natija=un/100+bir/100;
print(natija,'tiyin');
```



10.20-masala. Tushirib qoldirilgan harf o‘rniga H harfni yozishni o‘rgatuvchi dastur tuzing.

```
matn=str(input("Matn kiriting:"));
b=str(input('Harf kiriting:'));
t=matn.replace(b,"");
print(t);
```



7.5. MUSTAQIL BAJARISH UCHUN TOPSHIRIQLAR

Topshiriq: 1) Quyidagi satrlarni PYTHON dasturlash tilida tuzing:

10.1-masala. Matndagi bir xil so‘zlar va ularning sonini aniqlovchi dastur tuzing.
10.2-masala. Berilgan matndagi YA va E harflar hamda bo‘sh joylar sonini aniqlang.
10.3-masala. ASC II jadvalidan katta lotin harflarni chiqaring..
10.4-masala. Matnda izlanayotgan xarf(belgi) necha marta uchrashini aniqlaydigan dastur tuzing.
10.5-masala. Berilgan matndagi hamma R harflarni olib tashlang.
10.6-masala. Matndagi eng qisqa so‘zni va uning o‘rnini aniqlovchi dastur tuzing.
10.7-masala. Berilgan matndagi E harflarni Y harflarga almashtiring.
10.8-masala. Berilgan matnni teskarisiga satrlab ekranga chiqaring**.**
10.9-masala. PYTHON tilidagi berilgan operatorlarning inglizcha - o‘zbekcha lug‘atini tuzing. Bunda inglizcha so‘z kiritilganda uning tarjimasi natija sifatida olinishini ta’minlang.
10.10-masala. Berilgan matnda faqat bir marta uchraydigan belgilarni (matnda qanday uchrasa, shu tartibda) ajrating**.**
10.11-masala. Matndagi i – so‘zni j – so‘z bilan almashtirish dasturini tuzing.
10.12-masala. Berilgan matnning o’ng tomonidan N ta belgilarni ajrating.
10.13-masala. Matndagi 2, 5, 6, 8 raqamlari qatnashgan so‘zlardan yangi matn xosil qilish dasturini va eng uzun so‘zni aniqlash dasturini tuzing**.**
10.14-masala. Topishmoq topish va uning javobini tahlil qilish dasturini tuzing**.**
10.15-masala. Berilgan so‘zlarni alfavit bo‘yicha tartiblash dasturini tuzing.
10.16-masala. Matn ikkita gapdan iborat. Matndagi gaplarni o‘rnini almashtiring.
10.17-masala. Berilgan natural sonning xona birliklarini ajratib yozing.
10.18-masala. Matndagi so‘z kurinishidagi sonni 1<N<99 suz bilan yozish dasturini tuzing.
10.19-masala. Matndagi necha foyz so‘z A va D xarflaridan boshlanadi (so‘zlar probellar bilan ajratilgan).
10.20-masala. Tushirib qoldirilgan harf o‘rniga H yoki X harflarni yozishni o‘rgatuvchi dastur tuzing.

Foydalaniladigan adabiyotlar ro‘yxati:

Asosiy adabiyotlar

1. Normurodov Ch.B. Mengliyev Sh.A. PHP7 dasturlash tili - O‘quv qo‘llanma – Termiz: “Xamidi xususiy firmasi”, 2020, 218 bet.
2. Vasilev A. N. Python na primerax. Prakticheskiy kurs po programmirovaniyu. — SPb. Nauka i Texnika, 2016. — 432 st.
3. Dj. Forse, P. Bisseks, U. CHan - Django. Razrabotka veb-prilojeniy na Python, Simvol-Plyus , 2010. – 456 st
4. Eric Matthes Python Crash Course, a hands-on, Project-Based, Introduction to Programming, ISBN-13: 978-1-59327-603-4, 2016, 562 p.

Qo‘shimcha adabiyotlar

1. Mirziyoev SH.M. Erkin va farovon, demokratik O‘zbekiston davlatini birgalikda barpo etamiz. O‘zbekiston Respublikasi Prezidenti lavozimiga kirishish tantanali marosimiga bag‘ishlangan Oliy Majlis palatalarining qo‘shma majlisidagi nutq, Toshkent, 2016. 56-b.
2. Mirziyoev SH.M. Tanqidiy tahlil, qat’iy tartib-intizom va shaxsiy javobgarlik – har bir rahbar faoliyatining kundalik qoidasi bo‘lishi kerak. Mamlakatimizni 2016 yilda ijtimoiy-iqtisodiy rivojlantirishning asosiy yakunlari va 2017 yilga mo‘ljallangan iqtisodiy dasturning eng muhim ustuvor yo‘nalishlariga bag‘ishlangan Vazirlar Mahkamasining kengaytirilganmajlisidagi ma’ruza, 2017 yil 14 yanvar’ –Toshkent, O‘zbekiston, 2017. 104-b.
3. Mirziyoev SH.M. Qonun ustuvorligi va inson manfaatlarini ta’minlash-yurt taraqqiyoti va xalq farovonligining garovi. O‘zbekiston Respublikasi Konstitutsiyasi qabul qilinganining 24 yilligiga bag‘ishlangan tantanali marosimdagi ma’ruza. 2016 yil 7 dekabrь- Toshkent, O‘zbekiston, 2017. 48-b.
4. Mirziyoev SH.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. Mazkur kitobdan O‘zbekiston Respublikasi Prezidenti SHavkat Mirziyoevning 2016 yil 1 noyabrdan 24 noyabrga qadar Qoraqalpog‘iston Respublikasi,viloyatlar va Toshkent shaxri saylovchilari vakillari bilan o‘tkazilgan saylovoldi uchrashuvlarida so‘zlagan nutqlari o‘rin olgan.-Toshkent, O‘zbekiston, 2017. 488-b.

Foydalanilgan internet saytlar

1. http://cppstudio.com –Python tilida programmalash bo‘yicha namunalar izoxlari bilan keltirilgan

2. http://cplusplus.com – Python tilida mavjud konstruksiyalar ta’rifi, ishlatish namunalari bilan keltirilgan.

3. http://www.compteacher.ru/programming - dasturlash bo‘yicha video darsliklar mavjud.

4. http://www.intuit.ru – internet universitet, dasturlash bo‘yicha yozma va video ma’ruzalar o‘qish, test sinovlaridan o‘tish va sertifikat olish imkoniyati mavjud.

5. http://www.metanit.com – Python dasturlash tillari, Django texnologiyasi bo‘yicha ma’ruzalar, dasturlar namunasini olish mumkin.

6. http://www.ziyonet.uz – dasturlash asoslari bo‘yicha referatlar topish mumkin.

7. http://www.tami.uz/kitob.php – dasturlash bo‘yicha yozma va video ma’ruzalar o‘qish va kitobni yuklab olish imkoniyati mavjud.

MUNDARIJA

Mengliyev Shaydulla Abdusalomovich 1981 yilda Surxondaryo viloyati Jarqo‘rg‘on tumanida tug‘ilgan. 2003 yilda Termiz davlat universiteti Amaliy matematika va informatika ta’lim yo‘nalishini tamomlagan. 2003-2005 yillarda - O‘zbekiston Milliy universitetida Amaliy matematika va axborot texnologiyalari mutaxassisligi bo‘yicha magistraturada o‘qigan. 2004-2006 yillar mobaynida dastlab Toshkent radiotexnika kasb-hunar kolleji o‘qituvchisi, so‘ngra Toshkent davlat iqtisodiyot universiteti “Oliy matematika” kafedrasi assistent - o‘qituvchisi lavozimida ishlagan.
Hozirda Termiz davlat universiteti Fizika-matematika fakulteti “Amaliy matematika va informatika” kafedrasida katta o‘qituvchi lavozimida faoliyat yuritmoqda. 2019-yilda texnika fanlari bo‘yicha falsafa doktori dissertasiyasini himoya qilgan. Texnika fanlari bo‘yicha falsafa doktori (PhD) dotsent.

Abdug‘aniev Otabek Allajonovich 1987 yilda Surxondaryo viloyati Boysun tumanida tug‘ilgan. 2009 yil Termiz davlat universitetida Amaliy matematika va informatika ixtisosligi buyicha bakalavr, 2013 yilda Amaliy matematika va axborot texnologiyalari mutaxassisligi bo‘yicha magistraturani tamomlagan. 2018-2020 yillar Termiz davlat universitetida tayanch doktoranturada o‘qigan. 2020 yilda Kadrlar malakasini oshirish va statistik tadqiqotlar institutida iqtisodiyot fanlari bo‘yicha falsafa doktori (PhD) dissertatsiyasini himoya qilgan.
Hozirda Termiz davlat universitetida Axborot texnologiyalari fakulteti dekani lavozimida faoliyat yuritmoqda

Shonazarov Soatmurot Qulmurodovich 1989 yilda Surxondaryo viloyati Angor tumanida tug‘ilgan. 2011 yilda Termiz davlat universiteti Amaliy matematika va informatika ta’lim yo’nalishini tamomlagan. 2011-2012-yillarda Angor tumani 20-umumiy o’rta ta’lim maktabida matamatika va informatika fanlari o’qituvchisi lavozimida ishlagan. 2012-2014-yillarda Termiz davlat universiteti Amaliy matematika va axborot texnologiyalari yo’nalishi bo’yicha magistraturani tugatgan.

2014-yildan hozirgi vaqtga qadar Termiz davlat universiteti Amaliy matematika va informatika kafedrasida o’qituvchi lavozimida ishlab kelmoqda.

To‘rayev Dilmurod Shokir o‘g‘li 1995 yilda Surxondaryo viloyati Termiz shahrida tug‘ilgan. 2020 yilda Termiz davlat universiteti Amaliy matematika va informatika ta’lim yo‘nalishini tamomlagan. 2020 yildan hozirgi vaqtgacha Termiz davlat universiteti Amaliy matematika (sohalar bo‘yicha) mutaxassisligi bo‘yicha magistraturada o‘qiyapti. 2020-2021 yillar mobaynida dastlab Termiz shahar 7-sonli umumiy o‘rta ta’lim maktabi o‘qituvchisi.

Hozirda Termiz davlat universiteti Amaliy matematika (sohalar bo‘yicha) mutaxassisligi bo‘yicha magistraturada 2-bosqichda talabasi.