

**«Ақмола облысы білім басқармасының
Бұланды ауданы бойынша білім бөлімі
Макинск қаласының мектеп-лицейі»
коммуналдық мемлекеттік мекемесі**

«Python бағдарламалау тіліндегі есептер»
(6 – 10 сынып оқушыларына арналған әдістемелік құрал)

Макинск қаласы, 2025 жыл

Әзірлеуші: Жалғас Алпамыс Базарбекұлы, «Python бағдарламалау тіліндегі есептер», 6 – 10 сынып оқушыларына арналған әдістемелік құрал, 2023 жыл, 31б

Аңдатпа

Оқушыларға арналған «**Python бағдарламалау тіліндегі есептер**» құрастырылды. Кітапшада бағдарламалау тілінің құрама командалары, символдар және сөзтіркестері, тізімдер, топтар, сөздіктер мен жұмыс, функциялар мен файлдар, жиымдар графикалық режим операторлары тақырыптары бойынша толық практикалық материалдар, есептер беріліп, оған сәйкес бағдарлама мысалдары келтірілген.

Ұсынылып отырған бағдарлама орта мектептердің оқушыларына арналған. Бағдарламалау тілін тереңдетіп оқитын сынып оқушыларына, информатика пәнінен олимпиадаға дайындауға оқушыларға көмекші құрал ретінде ұсынылады.

Мазмұны

Python бағдарламалау тіліне кіріспе.....	3
Python бағдарламалау негіздері. Алфавит.....	5
Сызықтық алгоритмді программалау.....	12
Шартты оператор.....	14
Жолдық типтермен жұмыс.....	20
Процедура мен функция.....	24
Көп мүшеліктер.....	27
Қорытынды.....	29
Пайдаланған әдебиеттер.....	30
Қосымшалар.....	31

Python бағдарламалау тіліне кіріспе

Python - бұл жоғары деңгейлі, жалпы мақсаттағы бағдарламалау тілі. Ол оқылуға жеңіл синтаксисімен, кең функционалдылығымен және көптеген кітапханалар мен құралдардың болуымен танымал. Python әсіресе веб-әзірлеме, деректерді талдау, жасанды интеллект, ғылыми есептеулер және автоматизация үшін кеңінен қолданылады.

Есептеу техникасы қауымдастығының зерттеуі бойынша, Python көптеген американдық колледждердің алғашқы бағдарламалау тілі ретінде жетекші орынға ие. Құрама Штаттарда 39 IT-факультетінің сауалнамасына сәйкес, ол Java-ны ауыстырып, жаңа бағдарламашылар назар аударатын алғашқы жалпыға арналған бағдарламалау тілі болды. 10 IT-факультетінің 8-і (80%) және 39-дан (69%) 27-і мамандықты оқудың бірінші сатысында Python тұр.

Тілдің өзі 23 жыл бұрын құрылған, бірақ жаңадан келгендерді бағдарламалауға үйрететін тамаша құрал ретінде ол жақында ғана ашылды. Raspberry Pi компаниясы Python-ға әсер етті, бірақ Python әлемнің жетекші компаниялары, атап айтқанда, Google, Yahoo және NASA пайдаланатындығын ұмытпау керек. Қазіргі уақытта Python 2008 жылдан бастап ең танымал бағдарламалау тілдері арасында тұрақты позицияға ие. Python-ды түсіну үшін кемінде ағылшын тілінің негізін білу қажет.

Ойындарды жасау – бұл хоббиді пайдалы сабаққа айналдырудың жолы. Python бағдарламалау тілі бүкіл әлемде кеңінен қолданылады. Python-да бағдарлама жасау оңай және жылдам, сондықтан бұл көптеген бағдарламалаушылардың сүйікті тілі. Python – веб бағдарламалау, компьютерлік қауіпсіздік, графика, машина жасау және басқа да көптеген салаларда қолданылады. Python-ды жақсы меңгерген адам әр түрлі IT-саласында үлкен сұранысқа ие. Стив Джобс: «Әр адам бағдарламалауды үйренуі керек, себебі ол бізді ойлауға үйретеді», - деген. Сондықтан да қазіргі таңда бағдарламалау тілдеріне аса мән берілуде. Осыған байланысты мектеп оқушыларын да IT-саласына қызығушылығын арттырып, бағдарламалау тілімен жұмыс жасау дағдыларын қалыптастыру қажеттігі туындайды. Қазіргі қоғамның көптеген саласында қолданылатын бағдарламалау тілін меңгертіп, туындаған мәселелерді өз бетімен шеше білу құзыреттілігін қалыптастыруымыз керек. Сол себепті 6–10сынып оқушыларына арнап «Python бағдарламалау тіліндегі есептер» жинағының бағдарламасын ұсынып отырмын.

Оқыту мақсаты: танымал Python тілінің негіздерін үйрету және бағдарламалау тілдеріне деген қызығушылықтарын ояту. Оқушылардың шығармашылық жеке тұлғасын қалыптастыру және ақыл-ойын, ойлау қабілетін, ынтасын дамыту. IT-саласындағы ғылыми жетістіктерден хабардар ету. Оқушыларға код көмегімен сурет салуды және анимациялауды үйреріп, өз ойындарын жасауға бағыттау.

Оқыту міндеттері:

- Python тілінің негіздерімен таныстыру;
- Python бағдарламалау тілінде код жазуға үйрету;
- Бағдарламалаушылық білімін қалыптастыру;
- Шығармашылық пен жұмыс жасауға үйрету, өз ойындарын құрастырып, IT саласына деген қызығушылықтарын ояту.

Күтілетін нәтиже: Оқушылар бағдарламадағы тәжірибелік жұмыстарды орындау арқылы бағдарламалау кезінде кездескен проблемаларды өз беттерімен шешіп, өздерінің білімдерін көрсетеді және алгоритмдерді өз сөздерімен түсіндіріп, сыныптастарына қателіктерді түзетуге көмектесетін болады. Бағдарламалау балаға логиканы, кеңінен ойлауды, зейін және қиялды дамытуға септігін тигізеді.

Жаңашылдығы: Python бағдарламалау ортасында оқушылар тақырып бойынша берілген есептердің программасын құрып, оны орындату, түзету, нәтижесін талдауды, олимпиада есептерін шығаруды жүзеге асырады.

Python-ның негізгі ерекшеліктері:

1. **Оқылуға жеңіл синтаксис:** Python синтаксисі қарапайым және интуитивті, бұл оны үйренуді және қолдануды жеңілдетеді. Кодты оқығанда, ол ағылшын тіліне ұқсас, сондықтан оны түсіну оңай.
2. **Динамикалық типтеу:** Python-да айнымалылардың типін жариялаудың қажеті жоқ. Типтер бағдарламаның орындалу кезінде анықталады. Бұл кодты жазуды тездетеді.
3. **Көп платформалы:** Python бағдарламалары Windows, macOS, Linux және басқа да операциялық жүйелерде жұмыс істей алады.
4. **Интерпретацияланатын:** Python-да код компиляцияланбайды, бірақ интерпретатор арқылы тікелей орындалады. Бұл бағдарламаларды іске қосуды тездетеді.
5. **Кең кітапханалар жиынтығы:** Python-да көптеген кітапханалар мен модульдер бар, бұл әртүрлі тапсырмаларды орындауды жеңілдетеді. Мысалы:
 - **NumPy** - сандық есептеулер үшін
 - **Pandas** - деректерді талдау үшін
 - **Matplotlib** және **Seaborn** - графикалық визуализация үшін
 - **Scikit-learn** - машиналық оқыту үшін
 - **Flask** және **Django** - веб-әзірлеме үшін
6. **Қолданудың кең ауқымы:** Python әртүрлі салаларда қолданылады:
 - Веб-әзірлеме (back-end, API)
 - Деректерді талдау, статистика және машиналық оқыту
 - Ғылыми есептеулер
 - Автоматизация және скрипттер жазу
 - Ойындар әзірлемесі
 - Жасанды интеллект және т.б.

Тақырыбы: Python бағдарламалау негіздері. Алфавит

Мақсаты: Python бағдарламалау негіздерін, бағдарламалау тілінің алфавитін қарастыру.

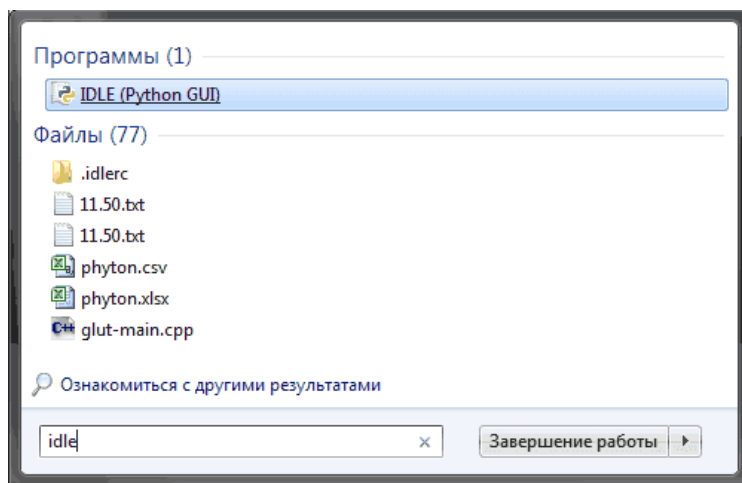
Жұмыс барысы:

Python-динамикалық күшті теру және жадының автоматты басқарылуы бар жоғары деңгейлі жалпы мақсатты бағдарламалау тілі. Әзірлеушінің өнімділігін, кодтың оқылуын және сапасын жақсартуға, сонымен қатар онда жазылған бағдарламалардың тасымалдануын қамтамасыз етуге бағытталған.

Python бағдарламасының мүмкіндіктері:

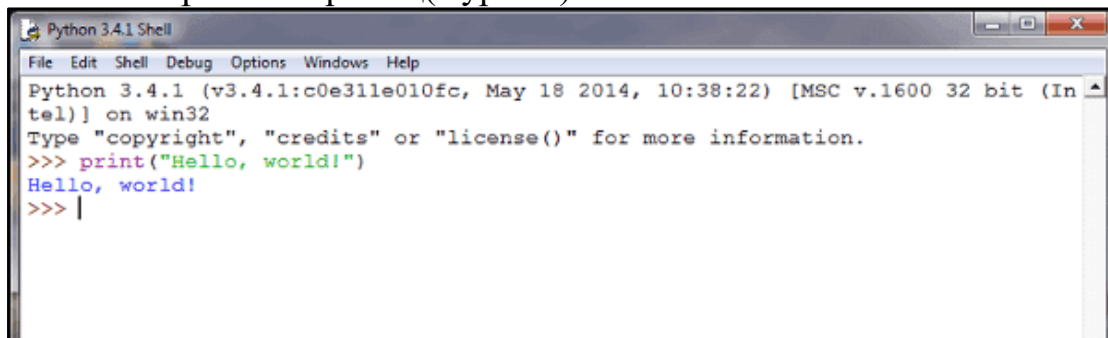
- xml/html файлдармен жұмыс;
- http сұраныстармен жұмыс;
- GUI (графикалық интерфейс);
- Веб-сценарийлерді құру;
- FTP-мен жұмыс;
- Сурет, аудио, видео файлдармен жұмыс;
- Робототехника;
- Математикалық және ғылыми есептеулермен жұмыс.

Python бағдарламасын іске қосу үшін Программы-> IDLE(Python)(Сурет 1).



Сурет 1. Python бағдарламасын іске қосу

Бағдарламаны іске қосқаннан соң, дәстүрлі түрде "hello world" бағдарламасын жүзеге асырайық(Сурет 2).



Сурет 2. "hello world" бағдарламасы

Интерактивті режимде жаңа файл қосу үшін File-> New командасын орындаймыз.

Тіл алфавиті- мәтінде қолданылатын белгілер жиынтығы.

Python бағдарламалау алфавиті:

1. а дан z мен А-дан Z-қа дейінгі латын әріптері;
2. 0, 1,2..9 араб цифрлары;
3. +, -, *, / арнайы символдар;
4. Қызметші сөздер: for, if, class, def және т.б.

Бақылау сұрақтары:

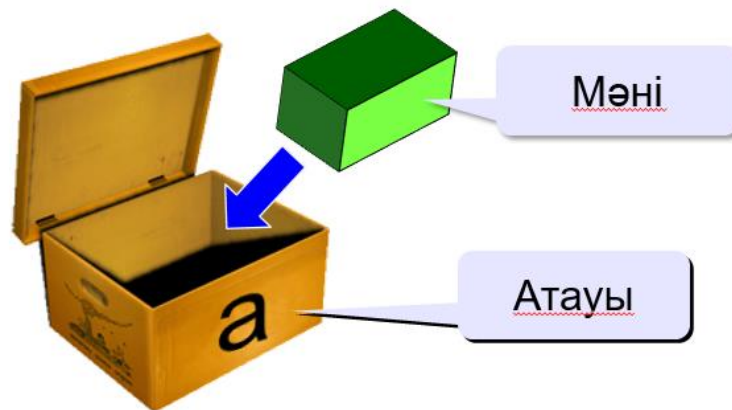
1. Python бағдарламасының мүмкіндіктерін атаңыз.
2. Python бағдарламалау тілінің алфавиті неден тұрады?

Айнымалылар. Айнымалылардың типтері.

Мақсаты: Бағдарламалаудағы айнымалы - бұл деректерді сақтауға арналған жадтың аймағын белгілейтін атау. Ол деректерді (мысалы, сандар, мәтін, тізімдер) сақтайды және бағдарламаның орындалу барысында оның мәнін өзгертуге болады.

Жұмыс барысы:

Айнымалы – атауы, типі мен мәні бар шама.



Сурет 3. Айнымалы

Айнымалыны сипаттау барысында төмендегі аталғандарды қолдануға болады:

- Латин әріптері(A-Z, a-z)
- Сандар;

Ал, орыс тілі әріптерін, жақша мен белгілерді(% , # , @ ..) қолдануға болмайды.

Мысалы:

a=5

c=12

Пернетақтадан мәндерді енгізу үшін

```
a = input()
```

қолданылады.

Ал, экранға шығару үшін **print** қызметші сөзі қолданылады.

```
print ( "2+2=?" )
```

```
print ( «Жауабы: 4" )
```

Екі мәнді бір жолға енгізу үшін

```
a, b = map ( int, input().split() )
```

қолданылады.

Айнымалылардың мынадай типтерін қарастырайық:

int # бүтін сандар- 5,12

float # бөлшек сандар- 4.12; 5.68

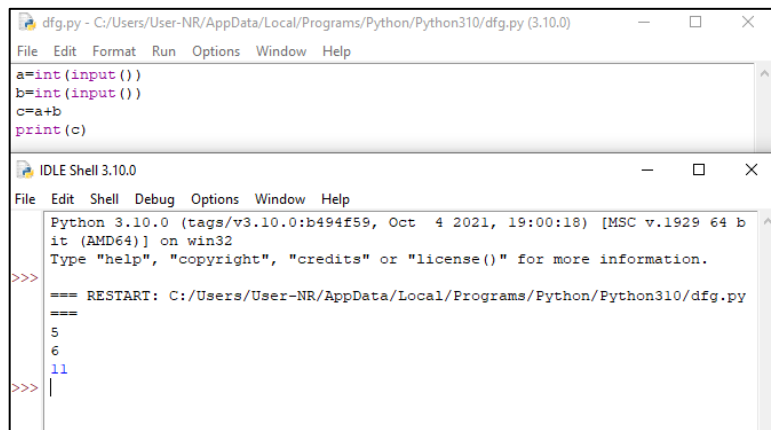
bool # логикалық мән- true, false

str # символдыжол- Nur-Sultan, Arman

Мысал (Python):

```
# Айнымалыларды жариялау және мәндерін беру
age = 30# Бүтінсан (int)
name = "John"# Мәтін (str)
height = 1.75# Нақтысан (float)
is_student = True# Логикалықмән (bool)
numbers = [1, 2, 3] # Тізім (list)
```

Программалық жүзеге асыру(Сурет 4).



```
dfg.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/dfg.py (3.10.0)
File Edit Format Run Options Window Help
a=int(input())
b=int(input())
c=a+b
print(c)

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/dfg.py ===
>>> 5
>>> 6
>>> 11
>>> |
```

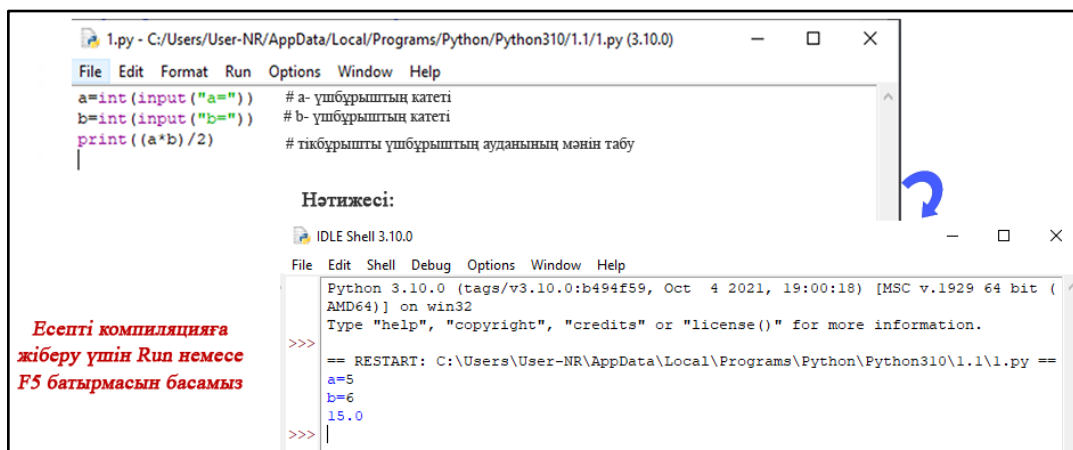
Сурет 4. Екі санның қосындысы

Бақылау сұрақтары:

1. Айнымалылардың қандай типтері бар?
2. Мәліметтерді пернетақта арқылы енгізу үшін қандай қызметші сөз қолданылады?

Практикалық жұмыс**Мысал №1**

Тікбұрышты үшбұрыштың катеттері берілген. Оның катеттері бойынша ауданын табыңыз.

Шығарылуы:


```
1.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/1.py (3.10.0)
File Edit Format Run Options Window Help
a=int(input("a=")) # a- үшбұрыштың катеті
b=int(input("b=")) # b- үшбұрыштың катеті
print((a*b)/2) # тікбұрышты үшбұрыштың ауданының мәнін табу

Нәтижесі:
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> == RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/1.py ==
>>> a=5
>>> b=6
>>> 15.0
>>> |
```

Есепті компиляцияға жіберу үшін Run немесе F5 батырмасын басамыз

Сурет 5. Мысалдың нәтижесі

Мысал №2

$$S = \sqrt{\cos 4y^2 + 7,151}$$

өрнегінің мәнін табыңыз.

Шығарылуы:

```

2.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2.py (3.10.0)
File Edit Format Run Options Window Help
import math
y=int(input("y="))
s=math.sqrt(math.cos(4*y*y)+7.151)
print(s)

Нәтижесі:
IDLE Shell 3.10.0
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2.py ==
y=1
2.5489912473636287
>>>

```

Бұл есепті шығару үшін математикалық math модулін іске қосамыз.

Сурет 6. Мысалдың нәтижесі

Мысал №3

$$R = \frac{\cos^3 y + 2^x d}{e^y + \ln(\sin^2 x + 7.4)}$$

өрнегінің мәнін табыңыз.

```

3.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/3.py (3.10.0)
File Edit Format Run Options Window Help
import math
x=int(input("x="))
y=int(input("y="))
d=int(input("d="))
r=(math.cos(y)**3 +2**x*d)/(math.exp(y)+math.log10(math.sin(x)**2+7.4))
print(r)

Нәтижесі:
IDLE Shell 3.10.0
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/3.py ==
x=1
y=1
d=1
0.5946745318933344
>>>

```

Сурет 7. Мысалдың нәтижесі

Өз бетінше жұмыс істеуге арналған тапсырмалар

А деңгейі

1. Екі сан берілен. Осы сандардың кубтарының арифметикалық ортасын және осы сандардың модулінің орташа геометриялық ортасын табыңыз.

$$F = \frac{\sqrt{|x| + \cos^3 x + z^4}}{\ln x - \arcsin(bx - a)}$$

2. өрнегінің мәнін табыңыз.

Б деңгейі

1. Радиусы белгілі шеңбердің ұзындығы мен ауданын табыңыз.
2. a,b,c қабырғалары бойынша үшбұрыштың ауданын табыңыз(Герон формуласы).

С деңгейі

1. Төртбұрыштың төбелерінің координаталары берілген. Координаталар бойынша оның периметрін анықтаңыз.
2. Дене $S = t^3 - 3t^2 + 2$ заңдылығы бойынша қозғалады. Белгілі бір уақыт аралығындағы дененің жылдамдығын табыңыз.

Тест сұрақтары

1. Python бағдарламалау тілінің алфавитінде қолдануға болмайтын белгілерді көрсетіңіз

- A. Араб цифрлары
- B. Арнайы символдар
- C. Қызметші сөздер
- D. Begin.. end

2. Python бағдарламалау ортасында айнымалыны белгілеу барысында төменде аталғандардың қайсысын қолдануға болмайды?

- A. Араб цифрлары
- B. жақшалар
- C. Қызметші сөздер
- D. Арнайы символдар

3. Пернетақтадан мәндерді енгізу үшін қай команданы қолдану қажет?

- A. Input()
- B. Print()
- C. Cin
- D. cout

4. Бос орындар бойынша жолдарды бөлу үшін қай команда қолданылады?

- A. Map
- B. Input()
- C. Split
- D. Goto()

5. Символды типті белгілеңіз

- A. Int
- B. Str
- C. Float
- D. Bool

6. Логикалық типті белгілеңіз

- A. Int
- B. Str
- C. Float
- D. Bool

7. Бүтін санның типін анықтаңыз

A.Int

B.Str

C.Float

D.Bool

8. Берілген $-3//4$ бүтін санды бөлу нәтижесі қандай?

A.1

B.-1

C.0

D.2+

9. $|x|$ мәнінің python бағдарламалау ортасында жазылуы

A.Sqrt(x)

B.Abs(x)

C.Abc(x)

D.Sqr(x)

10. $a\%b$ өрнегінің мағынасы

A. a санын b санға бөлу

B. a санын b санына қалдықсыз бөлу

C. a санының b дәрежесі

D. a санын b санына бөлу барысында бүтін бөлікті алу

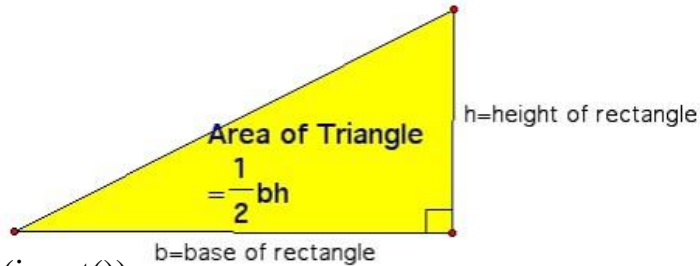
Бақылау сұрақтары:

1. Санды санға бөлгенде бүтін бөлікті анықтайтын функцияны атаңыз. Мысал келтіріңіз.
2. python бағдарламалау ортасында қолданылатын математикалық функцияларды атаңыз.

Сызықтық алгоритмді программалау

Есеп №1

Екі аяғының ұзындықтарын тік бұрышты үшбұрышта оқып, оның ауданын көрсететін программа жазыңыз. Әр сан бөлек жолға жазылады.



```
b = int(input())
h = int(input())
print(b*h/2)
```

Есеп №2 «Алманы бөлу».

п мектеп оқушылары k алманы бірдей бөледі, бөлінбейтін алдық себетте қалады. Әр оқушы қанша алма алады? Себеттен неше алма қалды? Бағдарлама кіріс ретінде және к сандарын алады және қажетті алма санын (екі сан) шығаруы керек.

```
n = int(input())
k = int(input())
print(k//n)
print(k % n)
```

Есеп №3 «Сандық сағат».

N саны берілген. Күн басталғаннан бері N минут өтті. Осы уақытта цифрлық сағат қанша сағат пен минутты көрсететінін анықтаңыз. Бағдарлама екі санды басып шығаруы керек: сағат саны (0-ден 23-ке дейін) және минут саны (0-ден 59-ға дейін). N саны тәуліктегі минуттар санынан көп болуы мүмкін екенін ескеріңіз.

```
n = int(input())
t = n // 60 % 24
m = n % 60
print(t, m)
```

Есеп №4 «Сәлем, Дастан!».

Сәлем сөзін, енгізілген есімді және тыныс белгілерін басып шығару арқылы келесі үлгі бойынша пайдаланушыға сәлем беретін бағдарлама жазыңыз:

```
name = input()
print('Hello, '+name+'!')
```

Есеп №5 «Келесі және алдыңғы».

Бүтін санды оқитын және көрсетілген мысалға ұқсас мәтін шығаратын бағдарлама жазыңыз (бос орындар маңызды!).

```
n = int(input())
print("The next number for the number " + str(n) + " is " + str(n + 1) + ".")
print("The previous number for the number " + str(n) + " is " + str(n - 1) + ".")
```

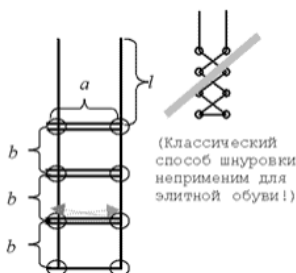
Есеп №6 «Мектеп парталары».

Мектеп үш жаңа математика сыныбын қабылдау туралы шешім қабылдады. Олардың математика сабақтары біруақытта өткізілетін болғандықтан, әр сыныпқа кеңсе бөліп, олардан жаңа парта сатып алу туралы шешім қабылданды. Әр партада екіден артық оқушы отыра алмайды. Үш сыныптың әрқайсысында оқушылардың саны белгілі. Барлық студенттерге жеткілікті болу үшін сізге қанша парта сатып алу керек? Бағдарлама кіріс ретінде үш натурал сандарды алады: үш кластың әрқайсысында оқушылардың саны.

```
a = int(input())
b = int(input())
c = int(input())
print(a // 2 + b // 2 + c // 2 + a % 2 + b % 2 + c % 2)
```

Есеп №7 «Бәтеңке баулары» .

Аяқ киім фабрикасы аяқ киімнің элиталық моделін шығара бастайды. Шілтерлік саңылаулар екі қатарға орналасады, жолдар арасындағы қашықтық a , ал қатардағы саңылаулар арасындағы қашықтық b . Әр қатардағы саңылаулардың саны N . Шілтерлеуді элиталық жолмен «жоғары, көлденеңінен басқа қатарға, жоғарыдан, көлденеңінен және төмен» жасау керек (суретті қараңыз). Сонымен қатар, шілтерді элиталық садақпен байлау үшін қашықтықтардың бос ұшының ұзындығы болуы керек. Бұл етіккешілтерлер қанша уақыт болуы керек?



```
a = int(input())
b = int(input())
L = int(input())
N = int(input())
print(2 * L + (2 * N - 1) * a + 2 * (N - 1) * b)
```

Тақырыбы: Шартты оператор

Мақсаты: Шартты оператордың қарапайым және күрделі құрылымын мысалдардың көмегімен қарастыру. Тақырыпқа байланысты есептер шығару

```
if шарт:  
    оператор1  
else:  
    оператор2
```

Мысалы. Егер де x оң мән болатын болса, онда $y=x^2+1$. Ал, теріс болса онда $y=-x$
Шешімі:

```
x = int(input())  
if x > 0:  
    print(x*2+1)  
else:  
    print(-x)
```

Шартты операторларға байланысты есептерде бір немесе бірнеше шарт болуы мүмкін. Сол шартты біріктіру мақсатында салыстыру операторлары қолданылады(Кесте 1):

Кесте 1. Салыстыру және логикалық операторлар

==	теңдік
!=	Тең емес
>	үлкен
<	кіші
>=	Үлкен немесе тең
<=	Кіші немесе тең
Логикалық операторлар	
and	және
or	немесе
not	емес

Мысал 2. Төмендегі берілген өрнекті шығарыңыз

$$y = \begin{cases} 0 < x < 5; x^2 + 5 \\ x \\ 3 \end{cases}$$

```
If x>0 and x<5:  
    print(x**2+5)  
else:  
    print(x/3)
```

егер де бірнеше шарт енгізу қажет болатын болса, онда қосымша elif блогын енгізуге болады, мысалы:

```
age = 18
if age >= 21:
    print("Қолжетімді")
elif age >= 18:
    print("Қолжетімділік жартылай мүмкін")
else:
    print("қолжетімсіз")
```

Мысалы: Тақ санды табу есебі

```
if n%2==1:
    Print('tak')
Else:
    Print('jup')
```

Мысалы: Үш санның максимумын табу

```
a = int(input())
b = int(input())
c = int(input())
if a > b:
if a > c:
    print(a)
else:
    print(c)
else:
if b > c:
    print(b)
else:
    print(c)
```

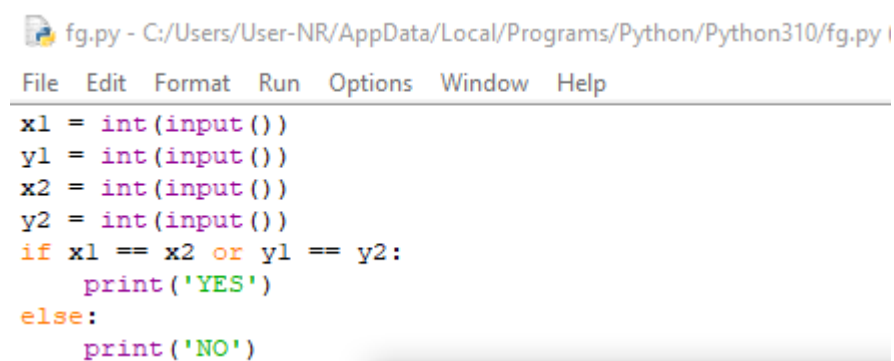
Бақылау сұрақтары:

1. Шартты оператор қандай жағдайда қолданылады?
2. Шартты опетордың жазылу конструкциясын көрсетіңіз.

Практикалық жұмыс

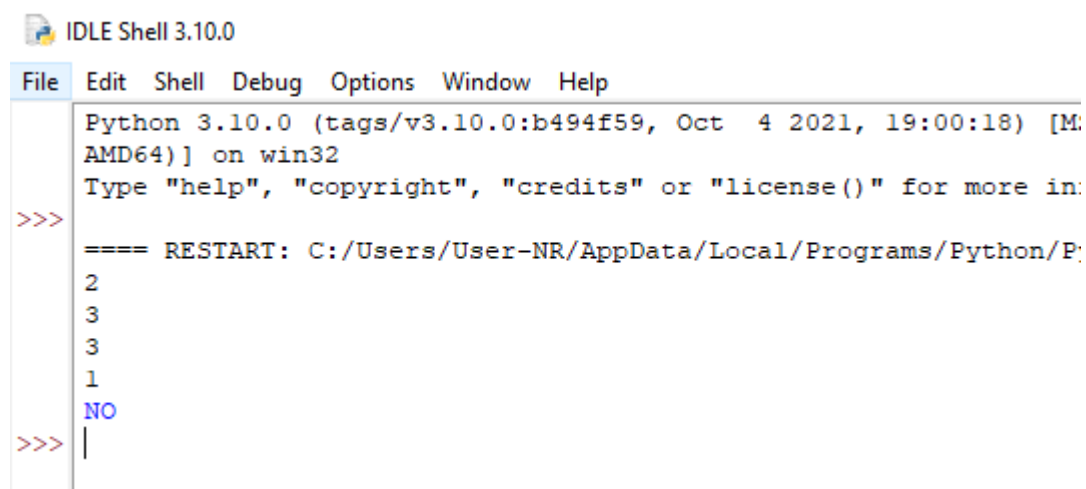
№1 мысал. Шахмат ойыны көлденең немесе тігінен қозғалады. Шахмат тақтасының екі түрлі квадратын ескере отырып, қарақшы бірінші шаршыдан екіншісіне бір қозғалыста жете алатынын анықтаңыз. Бағдарлама баған нөмірі мен жол нөмірін көрсете отырып, әрқайсысы 1 -ден 8 -ге дейінгі төрт санды алады, алдымен бірінші ұяшыққа, содан кейін екінші ұяшыққа. Бағдарлама бірінші ұяшықтан екінші ұяшыққа қозғалу арқылы кіру мүмкін болса, ИӘ басып шығаруы керек, әйтпесе ЖОҚ.

Шығарылуы:



```
fg.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/fg.py
File Edit Format Run Options Window Help
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if x1 == x2 or y1 == y2:
    print('YES')
else:
    print('NO')
```

Нәтижесі:



```
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [M:
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more in:
>>>
==== RESTART: C:/Users/User-NR/AppData/Local/Programs/P:
2
3
3
1
NO
>>> |
```

№2 мысал.

Шахмат патшасы көлденең, тігінен және диагональ бойынша қозғалады, бірақ тек 1 шаршы. Шахмат тақтасының екі түрлі квадратын ескере отырып, патшаның бірінші шаршыдан екіншісіне бір қимылмен жете алатынын анықтаңыз. Бағдарлама баған нөмірі мен жол нөмірін көрсете отырып, әрқайсысы 1 -ден 8 -ге дейінгі төрт санды алады, алдымен бірінші ұяшыққа, содан кейін екінші ұяшыққа. Егер патшаның қозғалысы бойынша бірінші ұяшықтан екіншісіне өту мүмкін болса, бағдарлама ИӘ шығаруы керек, әйтпесе ЖОҚ.

Шешімі:

```
*fg.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/f
File Edit Format Run Options Window Help
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if abs(x1 - x2) <= 1 and abs(y1 - y2) <= 1:
    print('YES')
else:
    print('NO')
```

Нәтижесі:

```
>>>
==== RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/fg.py =
5
3
2
3
NO
>>> |
```

№3 мысал. Шахмат пілі диагональ бойынша қозғалады. Шахмат тақтасының екі түрлі квадратын ескере отырып, піл бірінші шаршыдан екіншісіне бір қозғалыста жете алатынын анықтаңыз.

Шешімі:

```
fg.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/fg.py (3.10.0)
File Edit Format Run Options Window Help
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if abs(x1 - x2) == abs(y1 - y2):
    print('YES')
else:
    print('NO')
|
```

Нәтижесі

```
>>>
==== RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/fg.py =
5
3
2
3
NO
>>> |
```

Өз бетінше жұмыс істеуге арналған тапсырмалар

A деңгейі

1. Натурал сан берілген. Берілген сан тақ немесе 9-ға аяқталатынын анықтау қажет.
2. N саны 5-ке бөлінбейтін, бірақ 2 мен 3-ке қалдықсыз бөлінетіндігін анықтайтын логикалық тұжырым жазу қажет.
3. Пернетақтадан енгізілген сан үш өлшемді болатынын анықтау қажет.

B деңгейі

1. Натурал сан мінсіз деп аталады, егер де ол өзінің бөлгіштерінің суммасына тең болатын болса. Мысалы, $6=1+2+3$. N саны берілген. N –нан кіші болатындай барлық мінсіз сандарды табу қажет.
2. Пернетақтадан үш сан енгізіңіз де олардың ішіндегі ең кіші санды анықтаңыз.

C деңгейі

1. Үш орынды сан берілген. Оның цифрларының қосындысы екі орынды сан болатынын және оның цифрларының көбейтіндісі үш орынды сан болатынын анықтау қажет.
2. Егер де символ немесе сан тура және кері оқығанда бірдей немесе тең болатын болса, оны палиндром деп атайды. Мысалы: қазақ, 121. Төрт орынды сан палиндром болатын немесе болмайтындығын анықтау қажет.

Тест тапсырмасы

1. `a = False`
`b = True`
`if a and b:`
`print(1)`
`else:`
`print(2)` нәтижесі қандай?

Жауабын жазыңыз: _____ (2)

2. `a = -14`

`b = -12`

`if a<=b:`

`print(a)`

`else:`

`print(b)` нәтижесі қандай?

Жауабын жазыңыз: _____ (-14)

3.

`a = 15`

`b = 8`

`28`

`if a==b:`

```
print(a)
else:
print(b)нәтижесі қандай?
Жауабын жазыңыз:_____ (8)
```

```
4.a = True
if not a:
print(1)
else:
print(2)нәтижесі қандай?
Жауабын жазыңыз:_____ (2)
```

```
5
a = False
b = False
if a or b:
print(1)
else:
print(2)
Жауабын жазыңыз:_____ (2)
```

Тақырыбы: Жолдық типтермен жұмыс

Мақсаты: Жолдық типтерде қолданылатын амалдармен, командалармен танысу.

Python-да жолдық типтер (string) - мәтіндік деректерді ұсыну үшін қолданылатын негізгі деректер типтерінің бірі. Жолдар тырнақшалардың ішінде жазылады (бірлік, қос немесе үштік). Len функциясын қолданып символдар санын (жол ұзындығы) білуге болады.

Python-да жолдармен жұмыс істеуге арналған көптеген әдістер бар. Міне, бірнеше маңызды әдістер:

- **.lower()**: Барлық таңбаларды кіші әріпке ауыстырады.
- **.upper()**: Барлық таңбаларды бас әріпке ауыстырады.
- **.strip()**: Жолдың басы мен соңындағы бос орындарды (пробелдерді) жояды.
- **.replace(old, new)**: Ескі жолды жаңа жолмен ауыстырады.
- **.split(separator)**: Жолды көрсетілген бөлгіш бойынша бөледі (тізімге айналдырады).
- **.join(iterable)**: Тізімдегі жолдарды бір жолға біріктіреді.
- **.find(substring)**: Жолдағы кіші жолдың алғашқы индексін қайтарады (немесе -1, егер табылмаса).
- **.count(substring)**: Жолдағы кіші жолдың санын қайтарады.
- **.startswith(prefix)**: Жол көрсетілген жолмен бастала ма, соны тексереді.
- **.endswith(suffix)**: Жол көрсетілген жолмен аяқтала ма, соны тексереді.

Мысал №1

Сізге бос орынмен бөлінген екі сөзден тұратын жол беріледі. Бұл сөздерді орнын ауыстырыңыз. Нәтижені жолға жазыңыз және алынған жолды басып шығарыңыз. Бұл мәселені шешкенде, циклдар мен if операторын қолдануға болмайды.

Жауабы

```
1 s = input()
2 first_word = s[:s.find(' ')]
3 second_word = s[s.find(' ') + 1:]
4 print(second_word + ' ' + first_word)
```

Мысал №2

Жол берілген. Жолдағы барлық 1 сандарды one сөзіне алмастырыңыз.

Жауабы:

```
1 print(input().replace('1', 'one'))
```

Мысалдар:

```
my_string = " Python Programming "  
print(my_string.lower()) # Нәтиже: python programming  
print(my_string.upper()) # Нәтиже: PYTHON PROGRAMMING  
print(my_string.strip()) # Нәтиже: Python Programming  
print(my_string.replace("Python", "Java")) # Нәтиже: Java Programming  
print(my_string.split()) # Нәтиже: ['Python', 'Programming']
```

```
words = ["Hello", "world!"]  
print(" ".join(words)) # Нәтиже: Hello world!
```

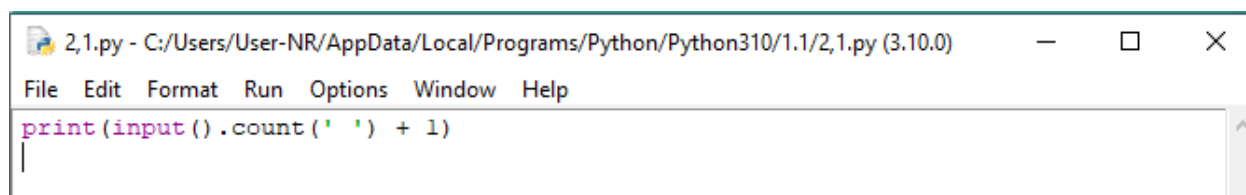
```
print(my_string.find("Programming")) # Нәтиже: 9  
print(my_string.count("m")) # Нәтиже: 2  
print(my_string.startswith(" ")) # Нәтиже: True  
print(my_string.endswith(" ")) # Нәтиже: True
```

Бақылау сұрақтары:

1. Жолдық типтерге қолданылатын әдістерді атаңыз.
2. Split әдісінің қызметін көрсетіңіз

Практикалық жұмыс

Мысал №1. Сөздерден тұратын жол берілген. Сөйлем неше сөзден тұратындығын анықтаңыз.



Мұндағы count – санауға арналған функция. Бұл жағдайда бос орындардың саны арқылы есептейді.

Нәтижесі:

```
>>> -  
= RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,1.py =  
Nur-Sultan arman kala  
3  
>>> |
```

Мысал №2

Екі сөзден тұратын жол берілген. Осы сөздердің орнын алмастырып, нәтижесін экранға шығарыңыз.

```
2,1.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,
File Edit Format Run Options Window Help
s = input()
first_word = s[:s.find(' ')]
second_word = s[s.find(' ') + 1:]
print(second_word + ' ' + first_word)
```

Мұндағы `first_word`, - жолдың бірінші жолы, `second_word`- жолдың екінші сөзі

Нәтижесі:

```
>>> = RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,1.py =
Nur-Sultan kalasy
kalasy Nur-Sultan
>>>
```

Мысал №3

Жол берілген. Бұл жолдағы барлық `h` символдарын `H` символына ауыстырызда экранға шығарыңыз.

```
2,1.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,1.py (3.10.0)
File Edit Format Run Options Window Help
s = input()
a = s[:s.find('h') + 1]
b = s[s.find('h') + 1:s.rfind('h')]
c = s[s.rfind('h'):]
s = a + b.replace('h', 'H') + c
print(s)
```

Нәтижесі

```
>>>
fdghghhgh gdfgfg
fdghgHHgh gdfgfg
>>>
```

Өз бетінше жұмыс жасауға арналған тапсырмалар

А деңгейі

1. Жол берілген. Берілген жолда қанша с символы бар екендігін анықтаңыз.
2. Нсимволы кем дегенде екі рет кездесетін жол берілген. Осы жолдағы бірінші және соңғы h әрпінің кездесуін жойыңыз.

В деңгейі

1. Символдар жолы берілген. Е әрпі осы жолда бар екендігін анықтаңыз.
2. Символдардың кейбір жолы берілген. Алғашқы нүктенің позициясын анықтаңыз. Жолдағы алғашқы символдың орналасуы 1-ден басталады.

С деңгейі

1. Пернетақтадан қандай да бір сөз енгізіледі. Осы сөз солдан оңға оңнан солға бірдей оқылатындығын анықтау қажет.
2. Жолдың ішіне енетін әр түрлі символдардың санын есептеу қажет.
3. Жолдағы символдарды код бойынша келесі символға ауыстырыңыз.

Тақырыбы: Процедура мен функция

Мақсаты: Процедура - бұл белгілі бір әрекетті орындайтын командалар тізбегі. Ол, әдетте, қандай да бір есептеуді жүргізбейді, бірақ деректерді өзгерту, шығару немесе басқа жанама әсерлер жасайды

Жұмыс барысы:

Python -да сіз өзіңіздің процедураларыңыз бен функцияларыңызды анықтай аласыз. Python процедуралары бір әрекеттер жиынтығы жиі қайталанғанда қолданылады. Процедуралар Python бағдарламасын құруға, оны көрнекі етуге мүмкіндік береді.

Python -да процедураны анықтау

```
def процедура атауы (аргумент1, аргумент2,..., аргументN):  
    процедуралармен орындалатын командалар
```

Функцияның берілуі

```
def функция атауы(аргумент1, аргумент2,..., аргументN):  
    функциямен орындалатын командалар  
    return мәні
```

Процедураның функциялардан айырмашылығы - Python функциясы белгілі бір мәнді қайтаруы керек, бірақ процедура жай командаларды орындайды.

Python -да функциялар жиынтығын енгізу үшін оның атын және оның барлық аргументтерін, үтірмен бөлінген жақшаларды көрсету керек, нүктелік белгілерден кейін процедуралық командаларды енгізу қажет.

Екі санның арифметикалық ортасын іздейтін және оны экранда шығаратын Python процедурасының мысалы.

```
def med(a, b):  
    return (a + b) / 2
```

```
print(med(a, b))
```

$ax + b = 0$ түріндегі теңдеудің шешімін табатын программаны процедура арқылы шығарайық.

```
def linear(a, b):  
    if (a == 0 and b == 0):  
        print("шексіз шешімдер саны")  
    if (a == 0 and b != 0):  
        print("шешімі жоқ")  
    if (a != 0):  
        sol = -b/a  
        print(sol)
```

```
a = int(input())
```

```
b = int(input())
```

```
linear(a, b)
```

```
medium(a, b)
```

Python -да жергілікті және жаһандық(ауқымды, глобальды) айнымалылар бар. Жергілікті айнымалылар тек көрсетілген функцияда қолданылады. Жаһандық айнымалылар бағдарламалануы керек және негізгі бағдарламада да, функцияларда да таныс. Әдепкі бойынша, Python берілген процедураларда айнымалыларды жергілікті етіп орнатады. Айнымалы команда жасау үшін процедураның басында команданы жазу керек

жаһандық айнымалы_атау

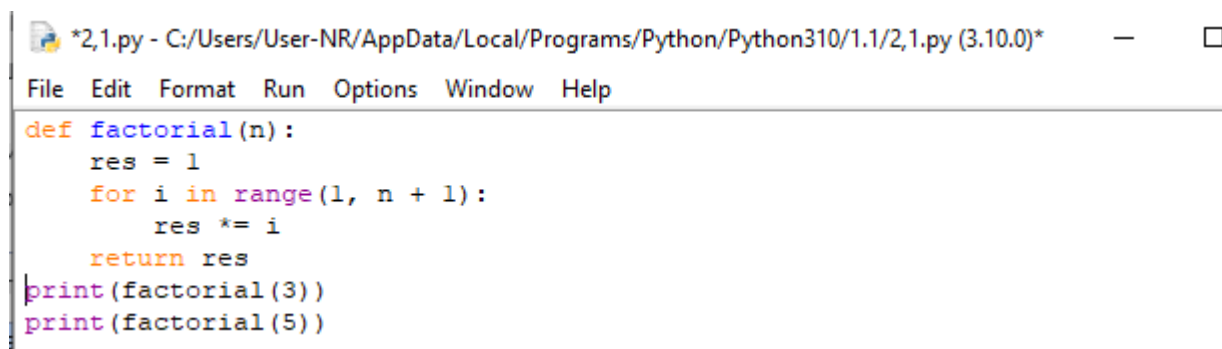
Бұл команда айнымалы мәнді жаһандық ретінде анықтайды. Бұл процедурада айнымалыны қолдануға болады, ал айнымалының мәні бүкіл бағдарлама бойынша өзгереді.

Бақылау сұрақтары:

1. Жергілікті және ауқымды айнымалы дегеніміз не?
2. Функция мен процедураның айырмашылығын мысалмен қарастырыңыз.
3. Функцияның жазылу синтаксисін көрсетіңіз.

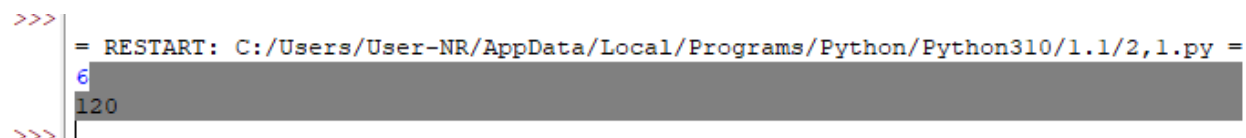
Практикалық жұмыс

Мысал №1. Санның факториалын анықтайтын программа құрайық. $6! = 120$



```
*2,1.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,1.py (3.10.0)*
File Edit Format Run Options Window Help
def factorial(n):
    res = 1
    for i in range(1, n + 1):
        res *= i
    return res
print(factorial(3))
print(factorial(5))
```

Нәтижесі:



```
>>>
= RESTART: C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,1.py =
6
120
>>>
```

Мысал №2. Үш санның максимумын табатын программа құрайық.

```
2,1.py - C:/Users/User-NR/AppData/Local/Programs/Python/Python310/1.1/2,1.py (3.10.0)
File Edit Format Run Options Window Help
def max(a, b):
    if a > b:
        return a
    else:
        return b

def max3(a, b, c):
    return max(max(a, b), c)

print(max3(3, 5, 4))
|
```

Нәтижесі

```
> |
= RESTART: C:/Users/User-NR/AppData/Local/Programs/Pyth
5
> |
```

Өз бетінше жұмыс істеуге арналған тапсырма

А деңгейі

1. Жай сандарды есептеу функциясын анықтау арқылы барлық екі таңбалы жай сандарды табыңыз.

2. Сандағы цифрларды анықтау функциясын жазу арқылы екі екі таңбалы a және b бүтін сандарының қосындысын табыңыз.

В деңгейі

1. Аудару функциясын қолдана отырып, ондық жүйеде берілген санды екілік жүйеге түрлендіру.

1.16 элементтен тұратын жол матрицасын ескере отырып, төменгі үшбұрышты матрицаны құру функциясын қолдана отырып, оның барлық элементтерінен төменгі үшбұрышты матрицаны құрыңыз.

С деңгейі

1. Оң және теріс екілік сан көрсетілді. Екілік санау жүйесінде сандарды қосу функциясын қолдана отырып, осы сандардың қосындысын есептеуге арналған программа құрыңыз

2. Сізге n әр түрлі натурал сан беріледі. Осы сандардың барлық ауыстыруларын басып шығарыңыз.

Тақырыбы: Көпмүшеліктер

Мақсаты: Көпмүшеліктердің қолдану мүмкіндіктерін практикалық түрде қарастыру

Жұмыс барысы:

Python көпмүшелік(жиынтық) - Python-да көпмүшеліктермен жұмыс істеудің бірнеше жолы бар. Еңкөпқолданылатынекіәдіс:

1. Коэффициенттертізімі (немесе кортеж)

арқылы: Бүләдіскөпмүшеліктіңкоэффициенттерінтізімнемесе кортеж ретіндесақтауғанегізделген.

Тізімдегіэлементтердәрежесітөменненжоғарығақарайорналасады.

2. **numpy.poly1d** класыарқылы: **numpy** кітапханасындакөпмүшеліктердіңғайлыба сқаруғаарналған **poly1d** класы бар.

Бүләдіскөпмүшеліктерменәртүрліамалдардыорындаудыжеңілдетеді.

Элементтер тізбектелген тізім ретінде сақталатын массивтерден айырмашылығы, жиынтықтарда элементтерді сақтау тәртібі анықталмаған (сонымен қатар жиынның элементтері тізімдегідей емес, ақылды алгоритмдерді қолдана отырып сақталады). Бұл жиынның барлық элементтері бойынша қайталаудан гөрі «элементтің жиынға жататынын тексеру» сияқты әрекеттерді жылдам орындауға мүмкіндік береді.

Жиын элементтері кез келген өзгермейтін деректер түрі болуы мүмкін: сандар, жолдар, кортеждер. Айнымалы деректер түрлері жиынның элементтері бола алмайды, атап айтқанда, сіз жиынның тізім элементін жасай алмайсыз (бірақ сіз кортеж жасай аласыз) немесе басқа жиын. Жиын элементтерінің өзгермеу талабы компьютердің жадында жиынтықты бейнелеу ерекшеліктерімен қойылады.

Жиын оның барлық элементтерін бұйра(фигурный) жақшаға тіркеу арқылы анықталады. Ерекшелік - бұл `set()` функциясының көмегімен жасалатын бос жиын. Егер орнатылған функция тізім ретінде, жол немесе корпус параметр ретінде берілсе, онда ол тізім элементтерінен, жолдан немесе кортежден тұратын жиынтығын қайтарады. Мысалға:

```
A = {1, 2, 3}
```

```
A = set('qwerty')
```

```
print(A)
```

Жиынның элементтер санын `len` функциясы көмегімен анықтауға болады.

For циклін қолдана отырып, жиынның барлық элементтерін алмастыра аласыз (анықталмаған тәртіпте!).

Жиынға элемент қосу үшін `add` функциясын қолдануға болады:

```
A = {1, 2, 3}
```

```
print(1 in A, 4 not in A)
```

```
A.add(4)
```

Жиынның элементтерін жою үшін discard және remove әдістері қолданылады.

Жиындарға қолданылатын әдістеріді қарастырайық(Кесте 3).

Кесте 3. Жиын әдістері

$A \cup B$ A.union(B)	A и B жиынын біріктіру
$A \cup B$ A.update(B)	B жиынының элементтерін A жиынына тіркейді
$A \cap B$ A.intersection(B)	A және B жиындарының қиылысуы
$A - B$ A.difference(B)	A және B жиындарының айырмашылықтарын көрсетеді
$A -= B$ A.difference_update(B)	B жиынына кіретін барлық элементтерді A элементінен жояды
$A \leq B$ A.issubset(B)	A жиыны B жиынының ішкі жиыны болса, true мәнді ұсынады
$A < B$	$A \leq B$ and $A \neq B$ мәніне ұқсас
$A > B$	$A \geq B$ and $A \neq B$ мәніне ұқсас

Бақылау сұрақтары:

1. Жиын дегеніміз не?
2. Жиынның тізімдерден қандай айырмашылығы бар?
3. Жиындарға қандай амалдар қолдануға болады?

Өз бетінше жұмыс істеуге арналған тапсырмалар

A деңгейі

1. Сандар тізімі берілген. Әр түрлі сандар саны қанша екендігін анықтаңыз.

B деңгейі

1. Екі сандар тізімі берілген. Бір жиынға және екінші жиынға кіретін элементтерді шығарыңыз.

C деңгейі

1. Белгілі бір сандар тізімі берілген. Егер де ол сан бұрын кездескен болса, онда ақиқат, ал кездеспеген болса жалған сөзін экранға шығарыңыз.

Қорытынды

Бағдарламалауға байланысты абстрактілі ойлауды дамыту заңдылықтарын талдай отырып, бұл процестің алғашқы шарты оқушылардың есептер тізбегін шешудің практикалық тәжірибесін үнемі жинақтап отыруы болып табылады, оның барысында олар кейде қиындықтарға тап болады және оларды жеңу үшін жаңа тәсілді қолдануға немесе бұрын қалыптасқан бірқатар ұғымдарды жалпылауға және қолдануға мәжбүр болады. Бұл ретте ұсынылып отырылған тәсілмен оқушыларға бағдарлауға берілген оқулықтағы кейбір тапсырмаларды қайта құрастыру арқылы ғылыми ұғымдардың барынша жылдам білу және түсіну нәтижесінен синкрет және комплекстік ойлау сатыларын қысқартуға жол ашылады. Абстрактілі ойлауды дамытудың жоғарыда аталған принциптері, бір қарағанда, айқын көрінеді, бірақ оны жүзеге асыру кезінде бірқатар себептерге байланысты көптеген проблемалар туындатады. Алғашқы және маңызды қиындықтар бағдарламалауды оқу курсының басында пайда болады, мұнда оқушыларға өте күрделі тапсырмалар беруге болмайды, бұл, біріншіден, оларға үлкен қызығушылық тудырады деп айтуға болмайды, екіншіден, олар бағдарламалау тілі синтаксисі және семантикасы тұрғысынан емес, бар білетін және түсінетін ұғымдардың көмегімен шешілетін жобаларды алдын-ала формалды қарастыра ала ма? Деген мәселені мұғалім ескере алуы керек. Үлкен көлемде код жазуды талап ететін тапсырмаларды құрамдас бөліктерге бөліп тапсырмалардың күрделігін азайту тәсілі бағдарламалауды үйрену тұрғысынан да, абстрактілі ойлауды дамыту тұрғысынан да бірден бірнеше артықшылықтар береді. Біріншіден, бұл оқушыларға жаңа бағдарламалау құралын үйрену әрқашан оның жаңа функционалдың пайда болу қажеттілігімен байланысты екенін түсінуге мүмкіндік береді, бұл қиындықты жеңуге мүмкіндік береді. Екіншіден, күрделі мәселені ойдағыдай шешу білу оқушылардың оқуға деген ынтасын арттырады. Сонымен, үшіншіден, жаңа тәсіл абстрактілі ойлауды қалыптастырып дамыту жолында біртіндеп алға жылжып, білім және дағды пирамидасын «аяқтауға» мүмкіндік береді.

Пайдаланылган әдебиеттер:

1. Берри Пол «Изучаем программирование на Python»- Москва, 2021г
2. Васильев А.Н. «Python на примерах»-СПб- Наука и техника, 2023г
3. Дмитрий Мусин«СамоучительPython Выпуск0.2»,2022г
4. Доусон М. Програмуем на Python. – СПб.: Питер, 2020. – 416 с.
5. Златопольский, Д. М. Основы программирования на языке Python / Д.М. Златопольский. - М.: ДМК Пресс, 2020. - 277 с.
6. Язык программирования Python. Сузи Р.А. Учебное пособие.–М.: Интернет Университет информационных технологий, 2021.–327с.
7. Марк Лутц. Программирование на Python. Тома 1 и 2, 4-е издание.–Пер. сангл. СПб.: Символ-Плюс, 2021.–992с.3
8. Саммерфилд М. Программирование на Python 3. Подробное руководство. Пер. сангл. Киселев А. – М.: Символ-Плюс, 2023.-608с.
9. Доусон М. Програмуем на Python.-СПб.: Питер, 2022.-416с.5
10. Гуриков, С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. - М.: Форум, 2020. - 991 с.
11. <https://www.python.org/>
12. <https://kk.wikipedia.org/wiki/Python>
13. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2020.
14. Гуриков, С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. - М.: Форум, 2022. - 991 с
15. Лутц, М. Программирование на Python, II том / М. Лутц. - СПб.: Символ-плюс, 2023. - 992 с.

Қосымшалар Арифметикалық өрнектер

$x+y$	Қосу
$x-y$	Айырма
$x*y$	Көбейту
x/y	Бөлу(нәтижеүнеміfloat типті болады) $11/5=2.2$ $10/5=2.0$
$x//y$	Бөлгендегі бүтін бөлігі (Екі санда бүтін болса, аргумент бүтін болады) $11//4=2$ $11.8//4=2.0$
$x\%y$	Бөлгендегі қалдық $11 \% 4=3$
$x**y$	X санының Y дәрежесі $2**3=8$
$abs(x)$	Модулі
$round(x)$	Дөңгелектеу $11.3 = 11$
$round(x,n)$	Үтірден кейінгі N санға дейін дөңгелектеу $ROUND(12.456123,3)=12.456$
$pow(x,y)$	$X**Y$ Жазбасының аналогы

Ескеру: +, -, *, ** амалдарының нәтижесі int типті болады, егер екі аргументте бүтін болса.

Тізбектегі сандармен қолданылатын функциялар

$\max(a, b, \dots)$	Үлкенін табу $\max([2,6,3,1]) = 6$
$\min(a, b, \dots)$	Кішісін табу $\max([2,6,3,1]) = 1$
$\max(seq)$	Үлкенін табу $\max([2,6,3,1]) = 6$
$\min(seq)$	Кішісін табу $\max([2,6,3,1]) = 1$
$\text{sum}(seq)$	Қосындысын табу $\max([2,6,3,1]) = 6$
$\text{sorted}(seq)$	Сұрыптау $([2,6,3,1]) = [1, 2, 3, 6]$

Math модулінің функциялары мен тұрақтылары

$\text{trunc}(x)$	Бүтінге қарай дөңгелектеу
$\text{sqrt}(x)$	Квадраттық түбірі
$\text{exp}(x)$	Экспоненті
$\log(x), \log_2(x), \log_{10}(x)$	Натуралды, екілік, ондық логаримдары
$\sin(x), \cos(x), \tan(x)$	Синусы, косинусы, тангенсі, x-радиан
$\text{asin}(x), \text{acos}(x), \text{atan}(x)$	Арксинусы, арккосинусы, арктангенсі
$\text{degrees}(x)$	Радианды градусқа ауыстыру
$\text{radians}(x)$	Градусты радианға ауыстыру
$\sinh(x), \cosh(x), \tanh(x)$	Гиперболдық синусы, косинусы, тангенсі
$\text{asinh}(x), \text{acosh}(x), \text{atanh}(x)$	Кері гиперболдық синусы, косинусы, тангенсі
$\text{hypot}(x,y)$	Катеттері x, y болатын үшбұрыштың гипотенузасы
$\text{factorial}(x)$	Факториалы
$\text{gamma}(x)$	Гамма-функция

pi	Пимәнінқайтарады
e	e санынқайтарады

Жолдық операциялар

S1 + S2	Жолдарды біріктіру
S1 *n	Жолды қайталау
len(S)	Функция. Жолдың ұзындығының
S[n]	Индекс бойынша сілтеме
S2inS1	Логикалық операция. S2 жолы S1 жолда бар екендігін анықтау
min(S)	Функция. S жолының кіші кодты символы
max(S)	Функция. S жолының ең үлкен кодты символы

Функциямен әдістердің қызметі

Функция немесе әдіс	Қызметі
S.find(str, [start],[end])	Жол ішіндегі жолды іздеу. Біріншісін көрсетеді немесе -1
S.index(str, [start],[end])	Жол ішіндегі жолды іздеу. Біріншісін көрсетеді немесе ValueError
S.replace(шаблон, замена)	Шаблонды өзгерту
S.split(символ)	Жолды бөлгіштер арқылы бөліп жазу
S.isdigit()	Жолда цифрды анықтау
S.isalpha()	Жолда әріпті анықтау
S.isalnum()	Жолда цифрды немесе әріпті анықтау
S.islower()	Жолда төменгі регистрлі символдарды анықтау
S.isupper()	Жолда жоғарғы регистрлі символдарды анықтау
S.isspace()	Жолда көрінбейтін символдарды анықтау (пробел, символ перевода страницы ('\f'), «новая строка» ('\n'), «перевод каретки» ('\r'), «горизонтальная табуляция» ('\t') и «вертикальная табуляция» ('\v'))
S.istitle()	Сөз бас әріппен басталатындығын анықтау
S.upper()	Жолды жоғарғы регистрге ауыстыру
S.lower()	Жолды төменгі регистрге ауыстыру
ord(символ)	Символдың ASCII коды
chr(число)	ASCII кодының символы
S.lstrip([chars])	Жолдың басындағы бос орынды жою (пробел)
S.rstrip([chars])	Жолдың соңындағы бос орынды жою (пробел)
S.strip([chars])	Жолдың басындағы және соңындағы бос орынды жою (пробел)

