

8 ЖОЛДАРМЕН ЖҰМЫС ІСТЕУ

8.1 Негізгі анықтамалар

Айта кету керек, біз бұрын Python бағдарламалау тілінде жолдармен жұмыс жасаудың алғашқы дағдыларын алдық. Деректерді енгізуге шақыру немесе бағдарламада жауап беру, сондай-ақ жолды аудару, кесте, жолды пішімдеу сияқты сұрақтар қарастырылады. Бұл тақырыпта біз жолдармен жұмыс істеуге арналған функциялар мен әдістермен танысамыз.

Python-дағы жол-бұл **str** класының нысаны. Бұрын тақырыптарда сандық деректермен жұмыс істеу үшін типтерді келтіру функциялары қолданылды, мысалы, **int**, бұл келесі мысалда көрсетілгендей:

```
ind=int(input(" \n Введите индекс элемента "))
```

Суретте жолдар нөлден индекстелетіні көрсетілген, яғни егер **stroke="python"** операторы болса, онда жолдағы бірінші таңба нөлге тең болады және оған сілтеме **stroka[0]** ретінде жазылады. Сонымен қатар, Python-дағы жол элементтеріне, теріс индекстерді көрсету арқылы қол жеткізуге болады, мысалы, **print(stroka[-6])** операторы **p** таңбасын экранға шығарады.

```
0  1  2  3  4  5
p  y  t  h  o  n
-6 -5 -4 -3 -2 -1
```

Сурет 103 – Жолдарды индекстеу

Бағдарламалау кезінде бір таңбаны екінші таңбамен салыстырмас бұрын оны 128 әріптік-цифрлық таңбаларды кодтауды қолдайтын ASCII (American Standard Code for Information Interchange ақпарат алмасудың американдық стандарттық коды) кестесі арқылы санға айналдыру қажет екені белгілі.

Нөлден басталатын базалық кестенің алғашқы 32 коды аппараттық жасаушыларға (ең алдымен, компьютерлер мен баспа құрылғыларын өндірушілерге) беріледі. Бұл аймақта кез-келген тілдік таңбаларға сәйкес келмейтін басқару кодтары деп аталады және сәйкесінше бұл кодтар экранда да, басып шығару құрылғысында да көрсетілмейді, бірақ біз тізімге енгізетін басқару функцияларын кестеде көрсетеміз.

Кесте 7 - Нөлден басталатын негізгі кестенің алғашқы 32 коды

Символ	Код	Клавиши	Значение
--------	-----	---------	----------

nul	0	Ctrl + @	Нуль
soh	1	Ctrl + A	Начало заголовка
stx	2	Ctrl + B	Начало текста
etx	3	Ctrl + C	Конец текста
eot	4	Ctrl + D	Конец передачи
enq	5	Ctrl + E	Запрос
ack	6	Ctrl + F	Подтверждение
bel	7	Ctrl + G	Сигнал (звонок)
bs	8	Ctrl + H	Забой (шаг назад)
ht	9	Ctrl + I	Горизонтальная табуляция
lf	10	Ctrl + J	Перевод строки
vt	11	Ctrl + K	Вертикальная табуляция
ff	12	Ctrl + L	Новая страница
cr	13	Ctrl + M	Возврат каретки
so	14	Ctrl + N	Выключить сдвиг
si	15	Ctrl + O	Включить сдвиг
dle	16	Ctrl + P	Ключ связи данных
dc1	17	Ctrl + Q	Управление устройством 1
dc2	18	Ctrl + R	Управление устройством 2
dc3	19	Ctrl + S	Управление устройством 3
dc4	20	Ctrl + T	Управление устройством 4
nak	21	Ctrl + U	Отрицательное подтверждение
syn	22	Ctrl + V	Синхронизация
etb	23	Ctrl + W	Конец передаваемого блока
can	24	Ctrl + X	Отказ
em	25	Ctrl + Y	Конец среды
sub	26	Ctrl + Z	Замена
esc	27	Ctrl + [Ключ
fs	28	Ctrl + \	Разделитель файлов
gs	29	Ctrl +]	Разделитель группы
rs	30	Ctrl + ^	Разделитель записей
us	31	Ctrl + _	Разделитель модулей

Кестелерді кодтаудың ұлттық стандарттарына кодтық кестенің халықаралық бөлігі өзгеріссіз енеді, ал екінші жартысында ұлттық алфавиттердің кодтары, жалған графикалық белгілер және кейбір математикалық белгілер бар.

32 кодтан 127 кодқа дейін ағылшын алфавитінің таңбаларының кодтары, тыныс белгілері, сандар, арифметикалық амалдар және кейбір қосалқы белгілер орналастырылған. Негізгі ASCII кодтау кестесі 8-ші кестеде көрсетілген.

sp	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
p	q	r	s	t	u	v	w	x	y	z	{		}	~	
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	

Сонымен, жолдармен жұмыс жасау кезінде "А" символы "a" символымен бірдей емес екенін білу керек. Жолдың ұзындығы тек компьютердің жедел жадының көлемімен шектеледі және жол элементіне жүгіну үшін оның индексын тік жақшада көрсету жеткілікті. Сол сияқты, тізім элементіне жүгіну бұрын қарастырылған.

Python-дағы жолдар өзгермейтін дәйектілік болып табылады және **for** операторымен циклды қолдана отырып өңделеді. Индекске жүгіну арқылы жол таңбасын өзгерту мүмкін емес екенін түсіну маңызды.

```
stroka="Python"
stroka[0]="p" #недопустимый оператор
```

Жолды өңдеу кезінде оған төмендегі тізімде көрсетілгендей тікелей циклде жүгінуге болады.

```
stroka="python"
for i in stroka:
    print(i, end=" ")
```

Бұл кодтың нәтижесі **p y t h o n** сөзін экранға шығару болады.

8.2 Символдармен жұмыс істеуге арналған функциялар

Жолдармен жұмыс істеуге арналған негізгі функцияларды қарастырайық.

1. *Len () функциясы.*

Жолдармен жұмыс істеу кезінде **Len()** функциясы пайдалы болуы мүмкін, оның мақсаты жолдың ұзындығын анықтау болып табылады. **Print**

операторында біз жолдың әр элементіне сілтеме жасаймыз, жолдың атын атаймыз және әр элементтің индексін тік жақшаға аламыз. Әрине, төмендегі кодтың нәтижесі **P y t h o n** сөзін экранға шығару болады.

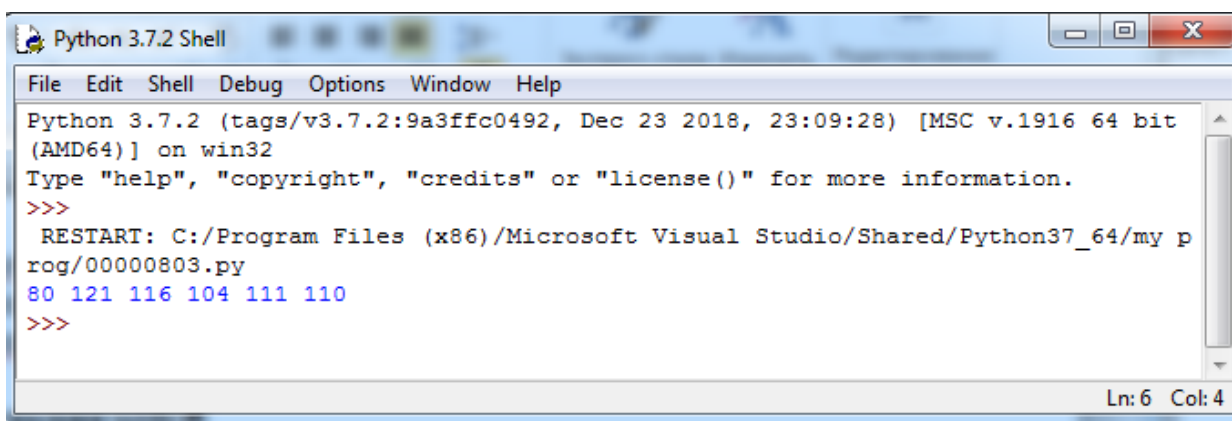
```
stroka="Python"
for i in range(len(stroka)):
    print(stroka[i], end=" ")
```

2. *Ord()* функциясы

Тағы бір функция - **ord()**, оның синтаксисі **ord("таңба")** көрсетілген таңбаның кодын қайтарады. Мысалы, төмендегі мысалда жолдың әр символына қолданған **ord()** функциясы **"Python"** жолының ASCII таңбалардың кодын қайтарады.

```
stroka="Python"
for i in range(len(stroka)):
    print(ord(stroka[i]), end=" ")
```

Бағдарлама жұмысының нәтижесі 104-ші суретте көрсетілген.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/00000803.py
80 121 116 104 111 110
>>>
```

Сурет 104 – Python сөзінің ASCII таңбалық кодтары шығарылды

3. *Chr()* функциясы

Chr(Сан) синтаксисі болатын функцияның әрекеті **ord()** функциясына тікелей қарама-қарсы, атап айтқанда таңбаның ASCII коды бойынша таңбаны қайтару. Осылайша, **ord()** функциясына қолданылған **chr()** функциясы, ол өз кезегінде ASCII таңбалық кодтарын қайтарады, төмендегі көрсетілген кодтағы **"Python"** жолының бастапқы түріне қайтарады.

```
stroka="Python"
for i in range(len(stroka)):
    print(chr(ord(stroka[i])), end=" ")
```

8.3 Жолдармен жұмыс істеу әдістері

Жолдармен жұмыс істеуге арналған негізгі әдістерді қарастырайық.

1. *Upper()* әдісі.

Stroka.upper() әдіс синтаксисі. Жолдың барлық таңбаларын жоғарғы регистрге түрлендіреді. Мысалы,

```
stroka="PythOn"  
newstr=stroka.upper()  
print(newstr)
```

2. *Lower()* әдісі.

Stroka.lower() әдіс синтаксисі.. Жолдың барлық таңбаларын төменгі регистрге түрлендіреді. Мысалы,

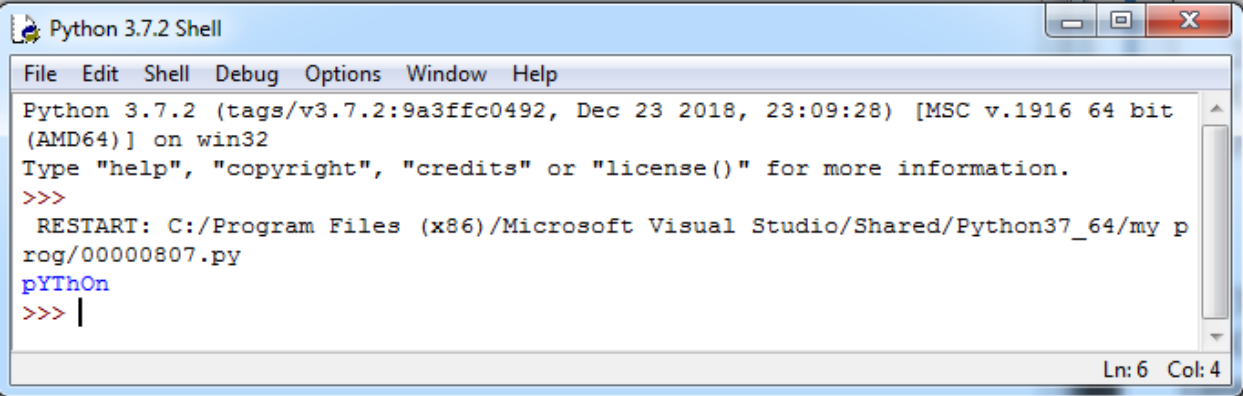
```
stroka="PytHon"  
newstr=stroka.lower()  
print(newstr)
```

3. *Swapcase()*әдісі.

Stroka.swapcase() әдіс синтаксисі. 105 – ші суретте көрсетілгендей, кіші әріптермен жазылған жолдың барлық таңбаларын жоғарғы және керісінше түрлендіреді. Мысалы,

Преобразует все символы строки, записанные в нижнем регистре - в верхний и наоборот как показано на рисунке 105. Например,

```
stroka="PytHoN"  
newstr=stroka.swapcase()  
print(newstr)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit  
(AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p  
rog/00000807.py  
pYThOn  
>>> |  
Ln: 6 Col: 4
```

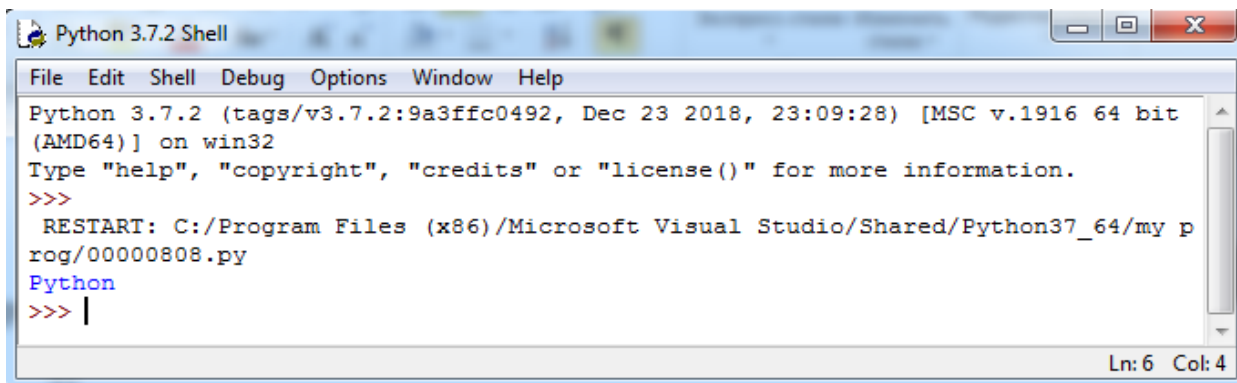
Сурет 105 – Swapcase() әдісінің жұмысы

4. *Capitalize()*әдісі.

Stroka.capitalize() әдіс синтаксисі. Жолдағы бірінші әріпті жоғарғы регистрге, ал қалғанын төменгі регистрге түрлендіреді. Мысалы,

```
stroka="pyTHoN"  
newstr=stroka.capitalize()
```

```
print(newstr)
```



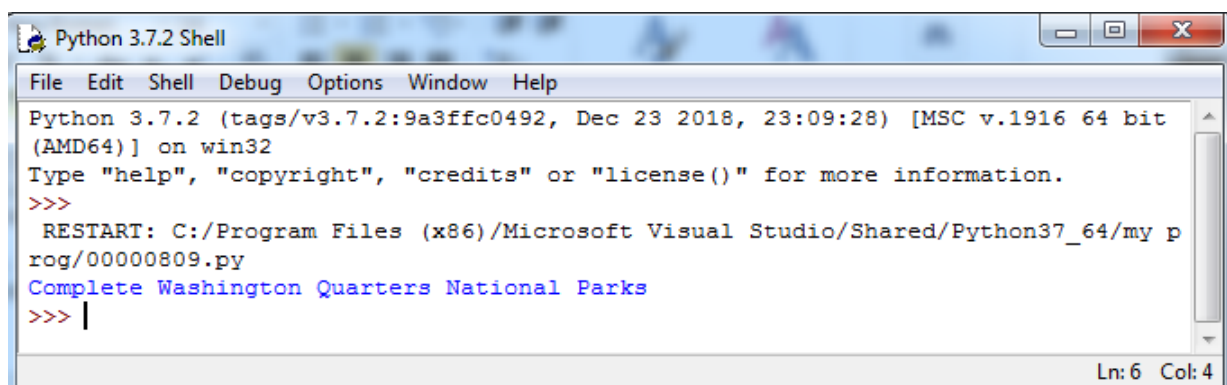
```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/00000808.py
Python
>>> |
```

Сурет 106 –Capitalize() әдісінің жұмысы

5. Title() әдісі.

Stroka.title() әдіс синтаксисы. 107-ші суретте көрсетілгендей жолдағы барлық бірінші әріптерді жоғарғы регистрге, ал қалғандары төменгі регистрге түрлендіреді. Мысалы,

```
stroka="complete washington quarTers national paRks"
newstr=stroka.title()
print(newstr)
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/00000809.py
Complete Washington Quarters National Parks
>>> |
```

Сурет 107 –Title() әдісінің жұмысы

6. Startswith() әдісі.

Stroka.startswith(podstroka) әдіс синтаксисі. **Stroka** жолының **podstroka**(жол үзіндісінен) көрсетілген ішкі жолдан басталатынын тексереді. Мысалы, Бұл мысалда "толық washington quarTers national paRks" жолының басында "com pl" ішкі жолының болу шарты тексеріледі. Бұл кодтың нәтижесі **true** мәні болады.

```
stroka="complete washington quarTers national paRks"
podstr="compl"
n=stroka.startswith(podstr)
print(n)
```

7. *Endswith()* әдісі .

Әдіс синтаксисі: **stroka.endswith(podstroka)**. **podstroka** ішкі жолымен көрсетілген **stroka** жолының аяқталғанын тексереді. Мысалы, "complete washington quarTers national paRks " жолының соңында " rks " ішкі жолының болу шарты тексеріледі. Бұл кодтың нәтижесі **False**(жалған) болады. Бұл мәтінді регистрге енгізудің сезімталдығын тағы бір рет көрсетеді, өйткені біздің жолымыз "rks" емес "Rks "ретінде аяқталады.

```
stroka="complete washington quarTers national paRks"  
podstr="rks"  
n=stroka.endswith(podstr)  
print(n)
```

8. *Replace()*әдісі .

Әдіс синтаксисі: **stroka.replace(old, new)**, мұнда **old** - ауыстыру үшін ішкі жол, **new** - жаңа ішкі жол. Әдіс **stroka** жолында **old** ішкі жолын жаңа ішкі жолмен табады және ауыстырады. Мысалы, осы мысалда stroka=stroka.replace("washington quarTers national paRks ", "Washington quarters National parks") келесі жолды "Complete Washington quarters National parks" қайтарады

```
stroka="Complete washington quarTers national paRks"  
stroka=stroka.replace("washington quarTers national paRks", "Washington  
quarters National parks")  
print(stroka)
```

9. *Rfind()*әдісі .

Әдіс синтаксисі: **stroke.rfind (podstr)**, мұнда **podstr** - ішкі жол. Әдіс ішкі жолдың соңғы позициясының орнын жолға қайтарады. Егер **ішкі жол табылмаса**, онда -1 мәні қайтарылады. **Podstr** параметрінен кейін бастапқы позицияны және ішкі жолды іздейтін жолдағы соңғы позицияларды көрсетуге болады. Бұл параметрлер міндетті емес, егер бастапқы позиция болмаса, іздеу жолдың басынан бастап жүзеге асырылады. Сонымен, **төмендегі мысалда "on"** ішкі жолының **соңғы кірісі 17** позициясында жазылады, өйткені біз іздеуде **8 таңбаны бастапқы позиция ретінде**, ал **соңғыны 30** ретінде көрсеттік. Егер мысалда **бастапқы және соңғы позициялар көрсетілмесе**, онда нәтиже **33** болады. Жол, нөлден индекстеледі, бос орындар мен тыныс белгілері де таңбалар болып табылады.

```
stroka="Complete Washington quarters National parks"  
stroka=stroka.rfind("on", 8, 30)  
print(stroka)
```

10 Find() әдісі .

Әдіс синтаксисі: **stroke.find (podstr)**, мұнда **podstr** - ішкі жол. Алдыңғы әдіске қарағанда, **find ()** әдісі жолдың бірінші ішкі жол позициясының орнын қайтарады. Егер **ішкі жол табылмаса**, онда -1 мәні қайтарылады. Тиісінше, егер сіз алдыңғы бағдарламаны өзгертсеңіз және **rfind()** әдісінің орнына **find ()** әдісін қолдансаңыз, сонымен қатар іздеудегі бастапқы және соңғы позицияларды алып тастасаңыз, онда " Complete Washington quarters National parks " жолына "on" ішкі жолының бірінші енгізуі 17-ші позицияда басталады.

```
stroka="Complete Washington quarters National parks"  
stroka=stroka.find("on")  
print(stroka)
```

11. Count() әдісі.

Әдіс синтаксисі: **stroke.count(podstr)**, мұнда **podstr** - ішкі жол. Бұл әдіс **substr** ішкі жолының пайда болу санын **stroka** жолына қайтарады. Осылайша, төмендегі мысалда **print** операторы экранға шығарған жауап **2** саны болады, өйткені "on" ішкі жолы " Complete Washington quarters National parks " жолына екі рет енеді.

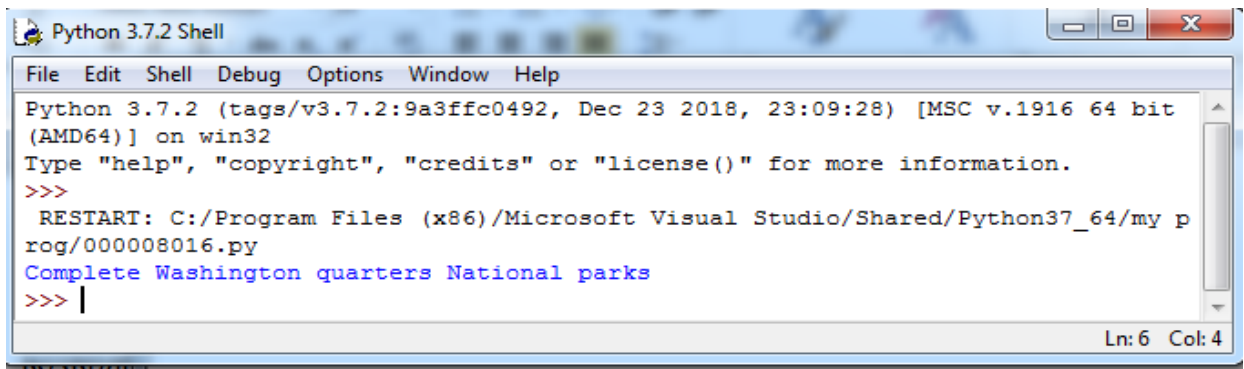
```
stroka="Complete Washington quarters National parks"  
stroka=stroka.count("on")  
print(stroka)
```

12. Strip() әдісі

Әдіс синтаксисі: **stroke.strip()**. Бұл әдіс **stroka** жолындағы бастапқы және соңғы бос орындарды жояды. Төмендегі мысалда бастапқы жолда жолдың басында және соңында бос орындар бар.

```
stroka=" Complete Washington quarters National parks "  
stroka=stroka.strip()  
print(stroka)
```

108-ші суретте бағдарламаның орындалу нәтижесі көрсетілген, онда **strip()** әдісі арқылы бос орындар бастапқы жолдан жойылады.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008016.py
Complete Washington quarters National parks
>>> |
```

Сурет 108 –strip() әдісінің жұмысы

13. Lstrip() u Rstrip()әдістері .

Их синтаксис и действие аналогичны рассмотренному методу **strip()**, с той разницей, что метод **lstrip()** удаляет символы пробела слева от начала исходной строки, а метод **rstrip()** - в конце исходной строки.

Отметим, что с помощью методов **strip()**, **rstrip()** и **lstrip()** можно удалять не только пробелы. Так, если в качестве параметра одного из методов указать символ или последовательность символов, то произойдет его (их) удаление. Например, в нижеследующем примере мы указали в качестве параметра в методе **strip()** символы "Complete ". Соответственно, метод возвратит исходную строку, но уже без указанных символов.

Олардың синтаксисі мен әрекеті қарастырылған **strip()** әдісіне ұқсас, айырмашылығы - strip() әдісі бастапқы жолдың басынан сол жақтағы бос орын таңбаларын жояды, ал **rstrip()** әдісі бастапқы жолдың соңында.

Strip(), **strip()** және **lstrip()** әдістерін қолдана отырып, бос орындарды ғана емес, жоюға болады. Сонымен, егер әдістердің бірінің параметрі ретінде таңбаны немесе таңбалар тізбегін көрсетсеңіз, онда ол(олар) жойылады. Мысалы, келесі мысалда біз **strip()** әдісіндегі параметр ретінде " Complete" таңбаларын көрсеттік. Тиісінше, әдіс бастапқы жолды қайтарады, бірақ көрсетілген таңбаларсыз.

```
stroka="Complete Washington quarters National parks  "
stroka=stroka.strip("Complete ")
print(stroka)
```

Ұқсас жағдай 109-суретте көрсетілген.

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008017.py
Washington quarters National parks
>>> |
Ln: 6 Col: 4

```

Сурет 108 –strip() әдісінің параметрмен жұмысы

14. Split() әдісі.

Әдіс синтаксисі: **stroka.split()**. Әдіс жолды ішкі жолдарға бөледі және оларды тізімге қосады. Егер бөлгіш параметр ретінде болса, онда жол көрсетілген бөлгішке сәйкес ішкі жолдарға бөлінеді. Ол болмаған жағдайда бос орын бөлгіш болып саналады. Төмендегі тізімде көрсетілген кодта **split()** әдісімен бастапқы жол тізімге айналдырылған, бөлгіш бос орын болып табылады. Бұдан әрі тізімнің үстінде тізімнің **remove()** әдісі орындалады, ол пайдаланушы пернетақтадан енгізген тізім элементін жояды.

```

stroka="America the Beautiful quarters: Complete Washington quarters
National parks"
spisok=stroka.split(" ")
print(spisok)
element=input("\n Введите элемент для удаления его из списка: ")
for i in spisok:
    if element in spisok:
        spisok.remove(element)
print("\n",spisok)

```

Бағдарлама жұмысының нәтижесі 109-суретте көрсетілген. Айта кету керек, тізімді құру кезінде бөлгіш бос орын болды. Тиісінше, бастапқы тізімде **"quarters"** тізімінің элементі оған **"quarters:"** тізімінің элементі сияқты бір рет кіреді. Жойылатын элементті сұраған кезде біз **"quarters"** енгіздік, сондықтан **"quarters:"** тізімінің элементі тізімде қалды және тек **"quarters"** тізімінен жойылды.

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008019.py
['America', 'the', 'Beautiful', 'quarters:', 'Complete', 'Washington', 'quarters', 'National', 'parks']
Введите элемент для удаления его из списка: quarters
['America', 'the', 'Beautiful', 'quarters:', 'Complete', 'Washington', 'National', 'parks']
>>>
Ln: 10 Col: 4

```

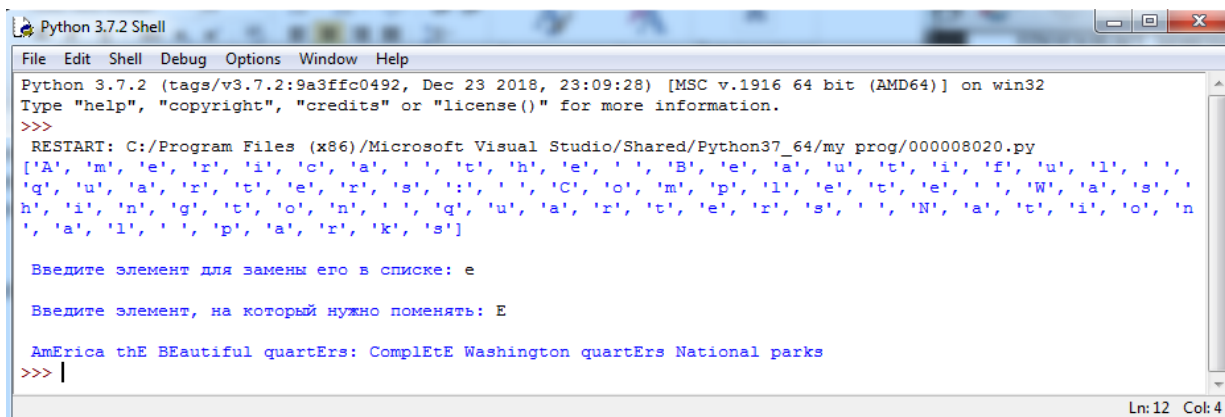
Сурет 109 – Тізімнен **"quarters"** элементін жою үшін **Split()** әдісінің тізімдік **remove()** әдісімен жұмысы

15. *Join()* әдісі .

Әдіс синтаксисі: **бөлгіш.join (spisok)**, онда **spisok** - тізім немесе кортеж. Жолды тізімге түрлендірудің тағы бір тәсілі-**list()** функциясын қолдану, ал кері түрлендіру **ljoin()** әдісімен жүзеге асырылады. Мұндай түрлендірулер жол элементін индекс бойынша өзгерту үшін қажет, өйткені жолдар бастапқыда өзгермейді.

Төмендегі мысалда **list()** функциясын қолдана отырып, бастапқы жолдан тізім жасалады. Бастапқы тізім элементін пернетақтадан енгізілген элементпен ауыстыруға тырысайық және ол үшін **input()** функциясын қолдана отырып, тиісті сұраныстарды орындаймыз. Біз **len()** функциясын қолдана отырып, тізімнің ұзындығын есептейміз және **for** операторының көмегімен тізімді қарауды ұйымдастырамыз. Егер тізімнің келесі элементі енгізілген элементке тең болса, онда элемент тізімінің бастапқы **element**-ті ауыстырылады **element1**-ге. Содан кейін, **join()** әдісін қолдана отырып, тізімді жолға айналдырып, оны экранға шығарыңыз (110-сурет).

```
stroka="America the Beautiful quarters: Complete Washington quarters  
National parks"  
spisok=list(stroka)  
print(spisok)  
element=input("\n Введите элемент для замены его в списке: ")  
element1=input("\n Введите элемент, на который нужно поменять: ")  
n=len(spisok)  
for i in range(0, n):  
    if spisok[i]==element:  
        spisok[i]=element1  
stroka1=""  
stroka1=""  
print("\n",stroka1)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008020.py  
['A', 'm', 'e', 'r', 'i', 'c', 'a', ' ', 't', 'h', 'e', ' ', 'B', 'e', 'a', 'u', 't', 'i', 'f', 'u', 'l', ' ', 'q', 'u', 'a', 'r', 't', 'e', 'r', 's', ':', ' ', 'C', 'o', 'm', 'p', 'l', 'e', 't', 'e', ' ', 'W', 'a', 's', 'h', 'i', 'n', 'g', 't', 'o', 'n', ' ', 'q', 'u', 'a', 'r', 't', 'e', 'r', 's', ' ', 'N', 'a', 't', 'i', 'o', 'n', 'a', 'l', ' ', 'p', 'a', 'r', 'k', 's']  
  
Введите элемент для замены его в списке: e  
  
Введите элемент, на который нужно поменять: E  
  
AmErica thE BEautiful quartErs: ComplEtE Washington quartErs National parks  
>>> |  
Ln: 12 Col: 4
```

Сурет 110 – "e" тізімінің элементін "E" - ге ауыстыру

8.4 Негізгі жолдарды өңдеу алгоритмдері

Мүмкіндіктер мен әдістерді қолдана отырып, жоғарыда келтірілген мысалдар Python-да жолдар мен таңбаларды өңдеу мүмкіндіктері туралы түсінік береді. Есептерді шешу кезінде Сіз жолдарды өңдеудің қарапайым әдістері мен жолдарды өңдеу әдістерін(оларды негізгі алгоритмдер деп атаймыз) қолдануды үйренуіңіз керек, олардың ішінде мыналарды бөліп көрсетуге болады:

1. Таңбалар санын анықтау.
2. Жолдағы таңбаларды ауыстыру.
3. Жолдағы таңбаны жою.
4. Жолға таңба қою.
5. Топқа жататын таңбаны талдау.
6. Жол айналымы.
7. Алфавиттік іріктеу.
8. Жолдардың кесінділері.

Бұл негізгі алгоритмдер олардың қолданылуын тек оқу мәселелерін шешу үшін ғана емес, сонымен қатар қарапайым мәтіндік редакторларды, мәтінді орналастыруға арналған бағдарламалық жасақтаманы және т. б. жасау үшін де қолдана алады.

Таңбалар санын анықтау.

Есеп 8.4.1. Пайдаланушы **stroka** бастапқы жолын және **simvol** таңбасын енгізеді. Берілген таңба бастапқы жолда қанша рет кездесетінін есептеңіз.

Шешімі. Бастапқыда **k** ұяшығын нөлдендіргеннен кейін, ол таңбалар есептегішінің рөлін атқарады, біз бастапқы жолдың барлық таңбаларын төменгі **lower()**. әдісімен кіші әріптерге түрлендіреміз. Содан кейін циклде **spisok[i]** жолының ағымдағы **simvol** ізделетін символмен салыстырылады. **I** цикл параметрі 0-ден (жолдың басы) жолдың соңына дейін өзгереді(бұл үшін **len(spisok)** функциясының мәні бар **n** ұяшығы жауап береді). Жолдағы табылған таңбалар санын есептеуді **k+=1** операторы қамтамасыз етеді.

Төменде мәселені шешуге жауап беретін бағдарлама коды берілген.

```
k=0
stroka=input("\n Введите строку: ")
stroka1=stroka.lower()
spisok=list(stroka1)
simvol=input("\n Введите символ для поиска его повторений в списке: ")
n=len(spisok)
for i in range(0, n):
    if spisok[i]==simvol:
        k+=1
print("\n", k)
```

Бағдарламаның нәтижесі 111-ші суретте көрсетілген.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008401.py

Введите строку: Определение количества символов o.

Введите символ для поиска его повторений в списке: o

5
>>> |
```

Сурет 111 – Жолдағы таңбалар санын анықтау бойынша бағдарлама жұмысының нәтижесі

Жолдағы таңбаларды ауыстыру.

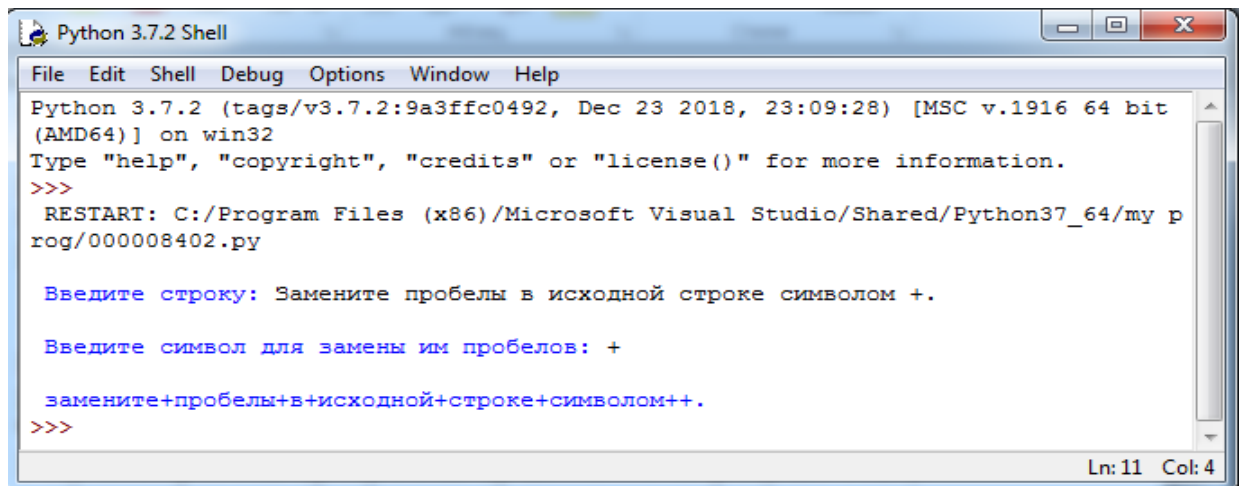
Есеп 8.4.2. Пайдаланушы **stroka** бастапқы жолын және **simvol** таңбасын енгізеді. Бастапқы жолдағы бос орындарды көрсетілген таңбамен ауыстырыңыз.

Шешімі . Бұрын **join()** әдісінің жұмысын түсіндірген кезде осындай есеп шешілді. Листингте өңделетін жол бекітілмеген, бірақ пернетақтадан енгізілген код берілген. Оны өңдеу әдістері өзгеріссіз қалды. Мұндай есепті **replace()** әдісін қолдана отырып шешуге болатындығын байқауға болады.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
stroka=input("\n Введите строку: ")
stroka1=stroka.lower()
spisok=list(stroka1)
simvol=input("\n Введите символ для замены им пробелов: ")
n=len(stroka1)
for i in range(0, n):
    if spisok[i]==" ":
        spisok[i]=simvol
stroka1="".join(spisok)
print("\n", stroka1)
```

Бағдарламаның нәтижесі 112-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008402.py

Введите строку: Замените пробелы в исходной строке символом +.

Введите символ для замены им пробелов: +

замените+пробелы+в+исходной+строке+символом++.
>>>
```

Сурет 112 – Бастапқы жолдағы бос орындарды + символымен ауыстыру бойынша бағдарлама жұмысының нәтижесі

Жолдағы таңбаларды жою.

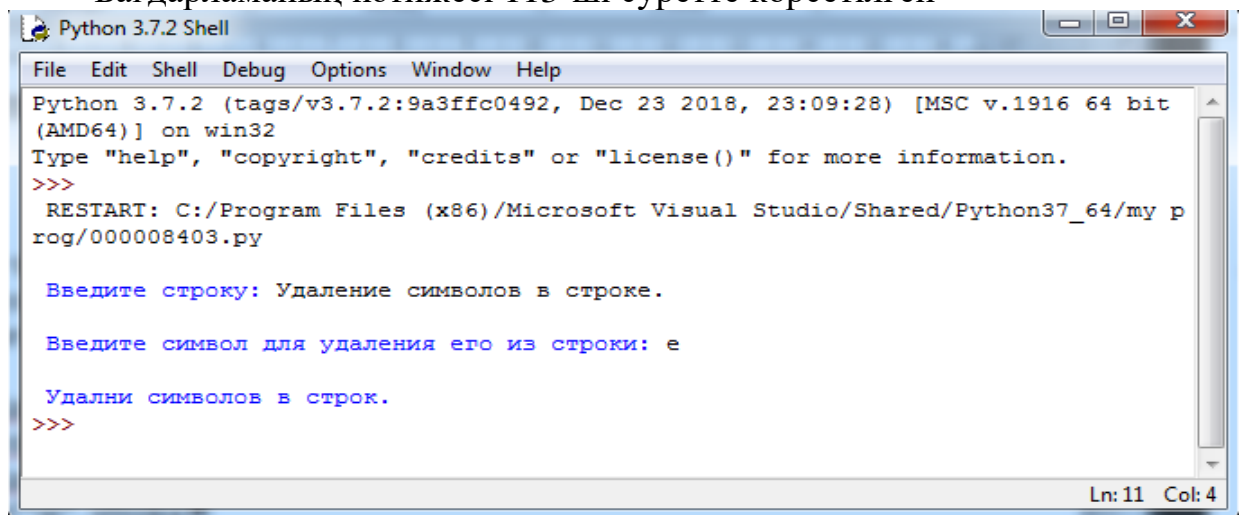
Есеп 8.4.3. Пайдаланушы **stroka** бастапқы жолын және **simvol** таңбасын енгізеді. Бастапқы жолда көрсетілген таңбаны жойыңыз.

Шешімі. Әдіс **replace()** әдісін қолдануға негізделген. Біз жойғымыз келетін таңбаны енгіземіз, содан кейін оны **replace()** әдісінде параметр ретінде қолданамыз. Әдісдегі екінші параметр бос тырнақшаларға тең болады.

Төменде мәселені шешуге жауап беретін бағдарлама коды берілген.

```
stroka=input("\n Введите строку: ")
simvol=input("\n Введите символ для удаления его из строки: ")
stroka=stroka.replace(simvol, "")
print("\n", stroka)
```

Бағдарламаның нәтижесі 113-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008403.py

Введите строку: Удаление символов в строке.

Введите символ для удаления его из строки: е

Удални символов в строк.
>>>
```

Сурет 113 – Бастапқы жолда енгізілген таңбаны жою бағдарламасының нәтижесі

Жолға таңба қою.

Есеп 8.4.4. Пайдаланушы бастапқы **stroka** жолын енгізеді. Пайдаланушы пернетақтадан енгізетін **simvol** таңбасынан кейін бастапқы жолға **simvol1** таңбасын қосу керек.

Шешімі Берілген позицияға ішкі жолды енгізу **spisok.insert (i+1, simvol1)**, операторының көмегімен жүзеге асырылады біз **list** функциясын қолдана отырып, жолды тізімге айналдырдық. **I** параметрін **for** операторымен циклде өзгерту жол бойынша жылжуды қамтамасыз етеді. Егер тізімнің келесі элементі табылған элементке тең болса (**if spisok[i]== simvol**), онда **insert** әдісі қолданылады. Бағдарламаның соңында, **join** әдісін қолдана отырып, кері түрлендіруді жасаймыз: тізім экранға шығарылатын жолға айналады. Жолдармен жұмыс істеу кезінде таңбаның жоғарғы немесе төменгі регистрде қандай күйде екенін ескеру қажет екенін есте ұстаған жөн.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
stroka=input("\n Введите строку: ")
print(stroka)
spisok=list(stroka)
simvol=input("\n Введите символ, после которого вставим введенный
ниже символ: ")
simvol1=input("\n Введите символ для вставки его в строку: ")
n=len(stroka)
k=0
for i in range(0, n):
    if spisok[i]==simvol:
        k+=1
for i in range(0, n+k):
    if spisok[i]==simvol:
        spisok.insert(i+1, simvol1)
stroka="".join(spisok)
print("\n", stroka)
```

Бағдарламаның нәтижесі 114-ші суретте көрсетілген

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008404.py

Введите строку: Вставка в строку символа а.
Вставка в строку символа а.

Введите символ, после которого вставим введенный ниже символ: а

Введите символ для вставки его в строку: _

Вста_вка_ в строку символа_ а_ .
>>> |
Ln: 14 Col: 4

```

Сурет 114 – Бастапқы жолға "а" символынан кейін енгізілген " _ " символын енгізу бойынша бағдарлама жұмысының нәтижесі

Топқа жататын таңбаны талдау.

Есеп 8.4.5. Пайдаланушы бастапқы **stroka** жолын енгізеді. Бастапқы жолдағы **symbol** және **simvol1** таңбаларының санын анықтаңыз. Пайдаланушы пернетақтадан **simvol1** және **simvol** таңбаларын енгізеді.

Шешімі . Бұл бағдарламада таңбалар жолын тізімге түрлендіріп, бастапқы жолда анықтағымыз келетін екі таңбаны енгіземіз. **For** операторымен циклде біз тізімнің әр элементін пайдаланушы енгізген таңбалармен салыстырамыз, егер шарт ақиқат болса, есептегіш бірге артады.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```

stroka=input("\n Введите строку: ")
print(stroka)
spisok=list(stroka)
simvol=input("\n Введите первый символ, количество которого
необходимо найти: ")
simvol1=input("\n Введите второй символ, количество которого
необходимо найти: ")
k1=0
k2=0
k=0
n=len(stroka)
for i in range(0, n):
    if spisok[i]==simvol:
        k+=1
        k1+=1
    if spisok[i]==simvol1:
        k+=1

```

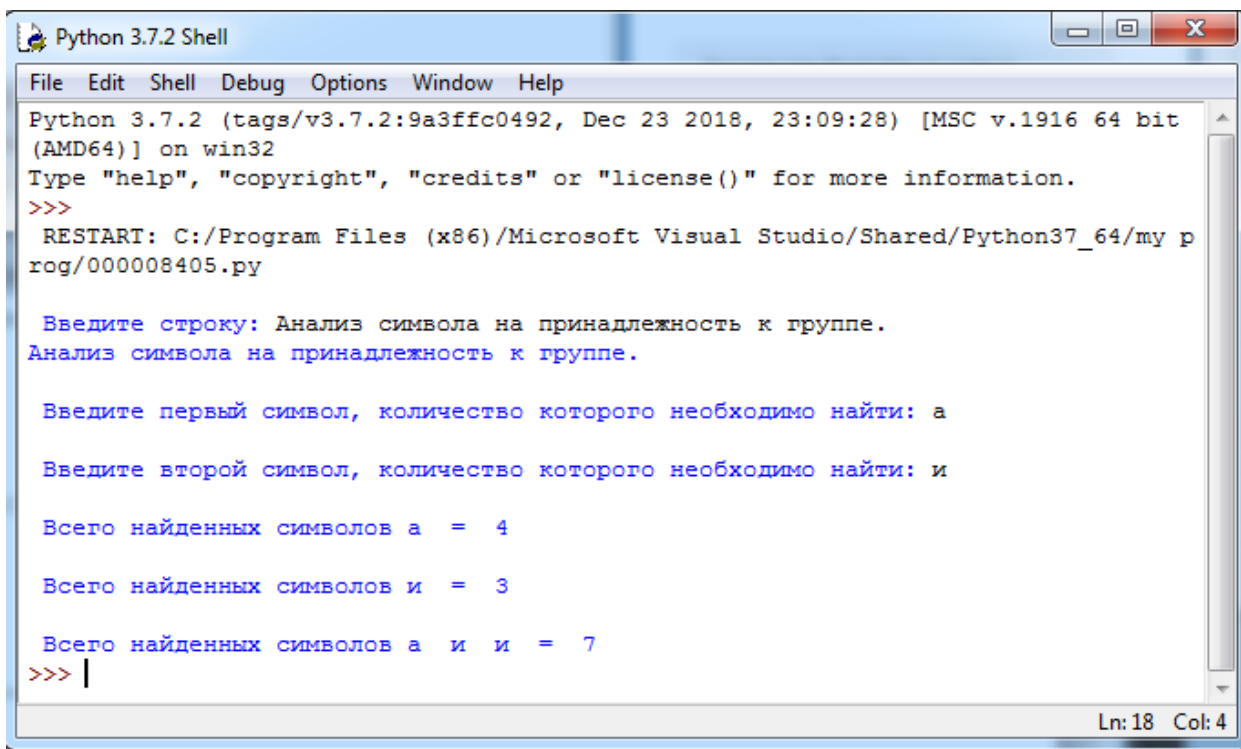


```

k2+=1
print("\n Всего найденных символов", simvol," = ", k1)
print("\n Всего найденных символов", simvol1," = ", k2)
print("\n Всего найденных символов", simvol, " и ", simvol1, " = ", k)

```

Бағдарламаның нәтижесі 115-ші суретте көрсетілген



```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008405.py

Введите строку: Анализ символа на принадлежность к группе.
Анализ символа на принадлежность к группе.

Введите первый символ, количество которого необходимо найти: а
Введите второй символ, количество которого необходимо найти: и

Всего найденных символов а = 4
Всего найденных символов и = 3
Всего найденных символов а и и = 7
>>> |
Ln:18 Col:4

```

Сурет 115 – Топқа жататын таңбаларды талдау бойынша бағдарлама жұмысының нәтижесі

Жол айналымы.

Есеп 8.4.6. Пайдаланушы бастапқы **stroka** жолын енгізеді. Жолды аудару керек, яғни таңбаларды кері ретпен жазу керек (соңғысы бірінші болады және керісінше).

Шешімі . Шешімі. Бұл мәселені шешкен кезде тізімдерді өңдеу үшін **reverse** әдісін қолдануға болады, бірақ бұл бағдарламаның алгоритмі берілген жолдан таңбаларды кезекпен таңдауға және оларды жаңа жолдың басына жылжытуға негізделген. **tmp** көмекші жолы бастапқыда бос. Цикл ұйымдастыру арқылы бастапқы жолдың символдарын біріншісінен соңғыға қарай қарастырамыз. Олардың әрқайсысы жинақталған жолдың басына қосылады, **tmp=stroka [i]+tmp** операторы көмегімен.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```

stroka=input("\n Введите строку: ")
print("\n", stroka)
tmp=""

```

```
n=len(stroka)
for i in range(0, n):
    tmp=stroka[i]+tmp
print("\n Строка наоборот: ", tmp)
```

Бағдарламаның нәтижесі 116-ші суретте көрсетілген

The screenshot shows a Python 3.7.2 Shell window. The title bar reads 'Python 3.7.2 Shell'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area contains the following output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008406.py

Введите строку: Обращение строки.

Обращение строки.

Строка наоборот: .икортс еинешарбО
>>>
```

The status bar at the bottom right indicates 'Ln: 11 Col: 4'.

Сурет 116 – Бағдарлама жұмысының нәтижесі

Алфавиттік таңдау.

Есеп 8.4.7. Алдын-ала белгіленген алфавит бар. Пайдаланушы бастапқы **stroka** жолын енгізеді. Алфавитте қолданылатын таңбаларды жолдан таңдап, оларды экранға шығарыңыз

Шешімі . Бұл мәселені шешкен кезде біз Python-да "тұрақты" сияқты ұғыммен танысамыз. Зерттелетін бағдарламалау тілінде бас әріптермен терілген айнымалы **тұрақты** деп аталады. Python-да тұрақтыларды қолдану ерекшелігі бар: көптеген бағдарламалау тілдерінен айырмашылығы, тұрақтыларды өзгертуге болады. Сондықтан тұрақты құрғаннан кейін пайдаланушы оның өзгермейтіндігін бақылауы керек.

Сонымен, бізде **ALFAVIT** тұрақтысында сақталатын алфавит деп аталатын белгілі бір таңбалар жиынтығы бар. Пайдаланушы жолды енгізеді және жолдың әр таңбасы үшін (**for letter in stroka:**) біз шартты тексереміз. Егер таңба алфавитте болса (**if letter in ALFAVIT:**), онда **tmp** айнымалысында біз **tmp+=letter** операторы арқылы алынған таңбаларды жинақтаймыз.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
ALFAVIT="abcdgh"
stroka=input("\n Введите строку: ")
print("\n", stroka)
tmp=""
for letter in stroka:
```

```

if letter in ALFAVIT:
    tmp+=letter
print(tmp)

```

Бағдарламаның нәтижесі 117-ші суретте көрсетілген

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008407.py

Введите строку: Complete Washington quarters National parks

Complete Washington quarters National parks
ahgaaaa
>>>
Ln: 10 Col: 4

```

Сурет 117 – Алфавиттік таңдау бағдарламасы жұмысының нәтижесі

Жолдардың кесінділері.

Есеп 8.4.8. Бастапқы жолдан берілген бастапқы және соңғы мәндер арасында орналасқан таңбаларды алыңыз.

Шешімі . Кортөждермен жұмыс жасау тақырыбын зерделеу кезінде "кесу" ұғымымен танысу болды және кортеждің кесілуі элементтердің алдына белгіленген бастапқы (**a**) және соңғы (**b**) позицияларының арасында орналасқан кортеж элементтерін шығару нәтижесінде алынған деп айтылды. Жол кесінділерімен жұмыс істеу механизмі кортеж кесінділерімен жұмыс істеу принциптеріне өте ұқсас. Кескінділерді жолдарға қолдана отырып, біз олардан кез-келген таңбаны немесе қатар орналасқан таңбалар тізбегін таңдай аламыз. Мұны істеу үшін бізге кесу шекараларын көрсететін бастапқы және соңғы позициялар қажет.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

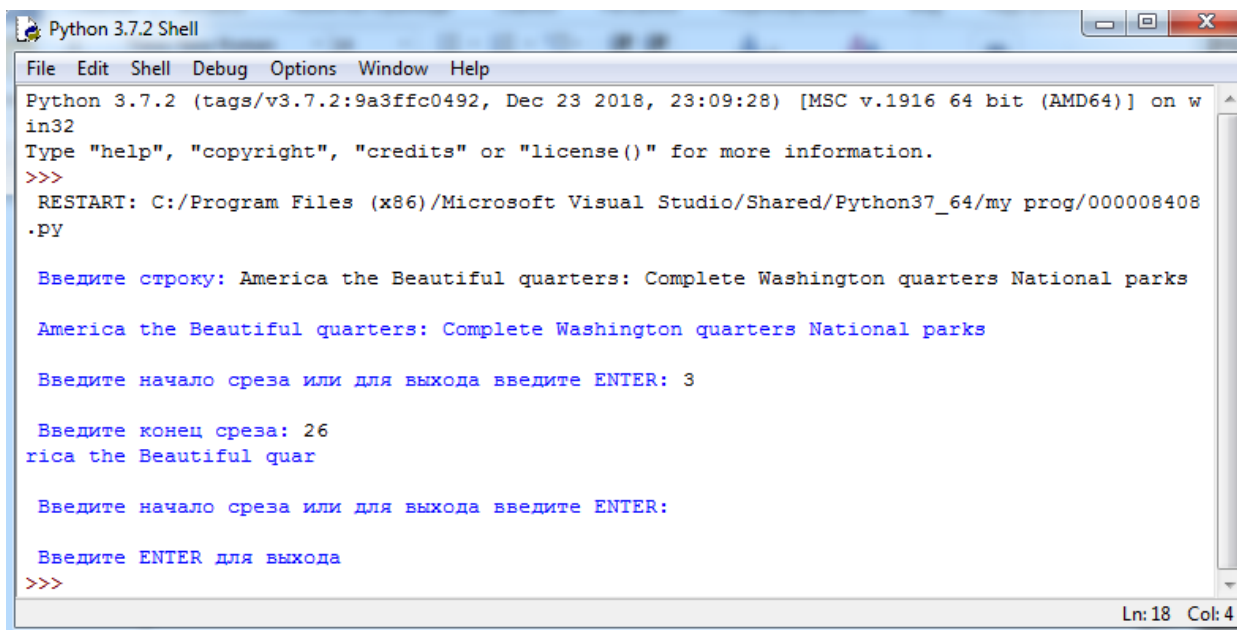
```

stroka=input("\n Введите строку: ")
print("\n", stroka)
flag=None
while flag!="":
    flag=input("\n Введите начало среза или для выхода введите ENTER:
")
    if flag:
        flag=int(flag)
        kon=input("\n Введите конец среза: ")
        kon=int(kon)

```

```
print(stroka[flag:kon])
input("\n Введите ENTER для выхода ")
```

Бағдарламаның нәтижесі 118-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on w
in32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008408
.PY

Введите строку: America the Beautiful quarters: Complete Washington quarters National parks

America the Beautiful quarters: Complete Washington quarters National parks

Введите начало среза или для выхода введите ENTER: 3

Введите конец среза: 26
rica the Beautiful quar

Введите начало среза или для выхода введите ENTER:

Введите ENTER для выхода
>>>
```

Сурет 118 – Жолды кесу бағдарламасының нәтижесі

8.5 есептерді шыаруға арналған мысалдар

Есеп 8.5.1. Пайдаланушы бірнеше сөзден тұратын **stroka** жолын енгізеді, олардың арасында бір немесе бірнеше бос орын бар. Ең ұзын сөздегі таңбалар санын табу керек.

Шешімі . For операторымен циклде біз келесі сөзге көшуді көрсететін жолдағы бос орынның болуын тексереміз. **Kol** айнымалысында сөздегі таңбалар саны сақталады. Тиісінше, сөздің максималды ұзындығын табу үшін біз бағдарламаға **maxim** сияқты айнымалыны енгіземіз, оған кішкене мән (мысалы, -1) тағайындағаннан кейін.

Сонымен, егер **kol** ұяшығының мәні **maxim** ұяшығының мәнінен асып кетсе, онда циклдің ағымдағы өтуінде жолдағы ең ұзын сөз болады. Сондықтан **maxim** ұяшығындағы циклден шыққан кезде жолдың ең ұзын сөзіндегі таңбалар саны сақталады.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
stroka=input("\n Введите строку: ")
kol=0
maxim=-1
n=len(stroka)
for i in range(0, n):
    if stroka[i]!=" ":
        kol=kol+1
```

```

if kol>maxim:
    maxim=kol
else:
    kol=0
print("\n Самое длинное слово имеет ", maxim, " символов")

```

Бағдарламаның нәтижесі 119-ші суретте көрсетілген

```

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my p
rog/000008501.py

Введите строку: State Quarters Proof Set PF 70 Ultra Cameo

Самое длинное слово имеет 8 символов
>>>
Ln:9 Col:3

```

Сурет 119 – Жолдағы ең үлкен сөздің ұзындығын табу бойынша бағдарлама жұмысының нәтижесі

Есеп 8.5.2. Пайдаланушы үтірмен бөлінген бірнеше сөзден тұратын **stroka** жолын енгізеді, бірақ кейбір жерлерде үтірден кейін бос орын жоқ. Бағдарламаның нәтижесі түзетілген жол болуы керек.

Шешімі . К айнымалысында жолдың ұзындығы сақталады. **While** операторымен циклде біз жол таңбаларын сұрыптай бастаймыз. Іздеу шарты ретінде келесі жол таңбасы "үтір" символына тең, ал үтірден кейінгі таңба бос орынға тең емес жағдайды тексереміз. Шарт ақиқат болғаннан кейін бос орынды қоямыз.

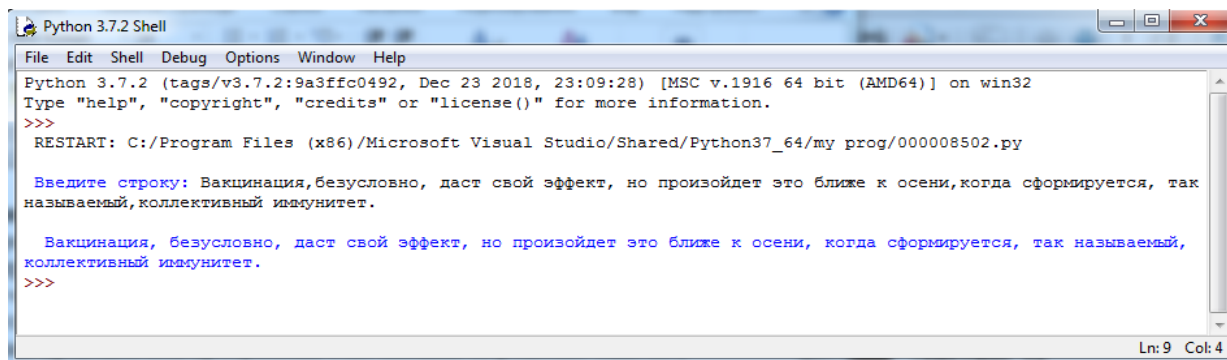
Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```

stroka=input("\n Введите строку: ")
n=len(stroka)
probel=" "
i=1
stroka1=stroka[0]
while i<n:
    stroka1=stroka1+stroka[i]
    if (stroka[i]==" ,"):
        if (stroka[i+1]!=" "):
            stroka1=stroka1+probel
    i=i+1
print("\n ", stroka1)

```

Бағдарламаның нәтижесі 120-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008502.py

Введите строку: Вакцинация, безусловно, даст свой эффект, но произойдет это ближе к осени, когда сформируется, так
называемый, коллективный иммунитет.

Вакцинация, безусловно, даст свой эффект, но произойдет это ближе к осени, когда сформируется, так называемый,
коллективный иммунитет.
>>>
```

Сурет 120 – Жолды түзету бойынша бағдарлама жұмысының нәтижесі

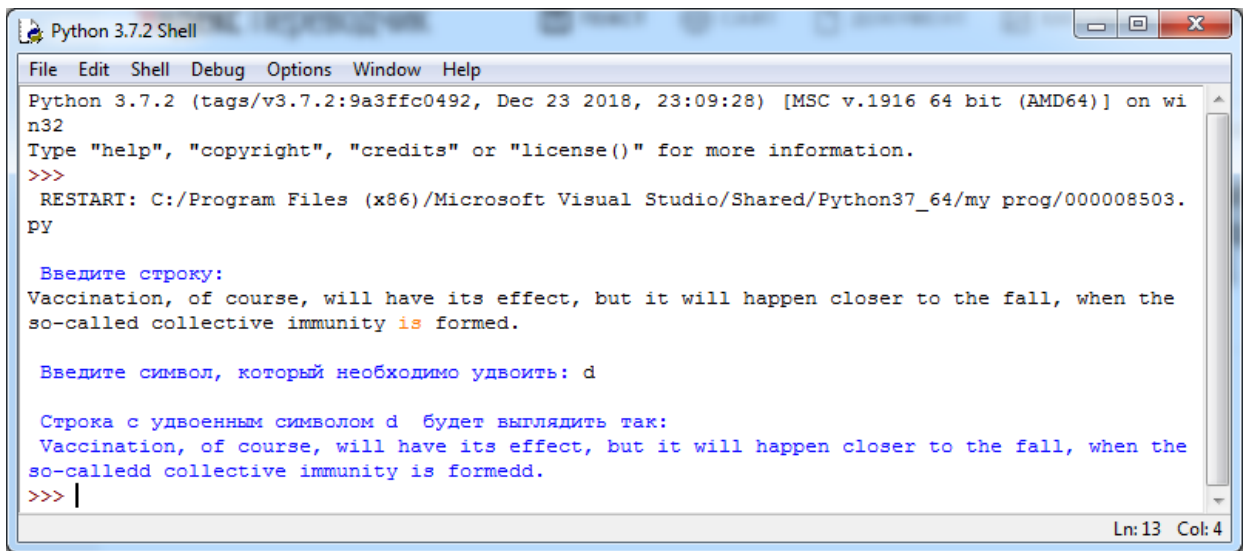
Есеп 8.5.3. Пайдаланушы жолда бірнеше сөзден және **simvol** таңбаларының бірінен тұратын **stroka** жолын енгізеді. Таңбаны екі есе артыру арқылы, алынған жолды шығару керек.

Шешімі. Бастапқы жолды енгізгеннен кейін **list** функциясын қолдана отырып, оны таңбалар тізіміне түрлендіреміз. **Zamena=simvol+simvol** операторы енгізілген символ таңбасын екі есе арттырады. **For** операторы бар циклде біз енгізілген таңбаның бастапқы жолдың таңбаларына теңдігін тексереміз. Егер шарт ақиқат болса, онда мұндай символ табылды және оны **spisok[i]=zamena** операторын орындау арқылы екі есе көбейтуге болады. Содан кейін тізімді **join()** әдісімен жолға түрлендіру және оны экранға шығару жүзеге асырылады.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
stroka=input("\n Введите строку: \n")
spisok=list(stroka)
simvol=input("\n Введите символ, который необходимо удвоить: ")
zamena=simvol+simvol
n=len(spisok)
for i in range (0, n):
    if spisok[i]==simvol:
        spisok[i]=zamena
strokazam=""
print("\n Строка с удвоенным символом", simvol, " будет выглядеть так:
\n", strokazam)
```

Бағдарламаның нәтижесі 121-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008503.py

Введите строку:
Vaccination, of course, will have its effect, but it will happen closer to the fall, when the so-called collective immunity is formed.

Введите символ, который необходимо удвоить: d

Строка с удвоенным символом d будет выглядеть так:
Vaccination, of course, will have its effect, but it will happen closer to the fall, when the so-called collective immunity is formedd.
>>> |
```

Сурет 121 – Символдарды екі еселеу бойынша бағдарлама жұмысының нәтижесі

Есеп 8.5.4. Пайдаланушы паролінің сенімділік дәрежесін тексеретін бағдарламаны құрыңыз, парольдің күрделілік критерийлері келесідей:

- сенімсіз-ұзындығы 6 таңбадан аз немесе ұзындығы 6-дан 8 таңбаға дейін, ал пароль тек бір регистрдегі сандардан немесе әріптерден тұрады;
- орташа-ұзындығы 6-дан 8 таңбаға дейін, сандарды және/немесе кіші және/немесе үлкен таңбаларды қамтуы керек немесе ұзындығы 8 таңбадан асуы керек, бірақ парольде бас әріптер, кіші әріптер немесе сандар жоқ;
- күрделі-пароль ұзындығы 8 таңбадан асады, оған бір уақытта бас әріптер, кіші әріптер мен сандар кіреді.

Шешімі . Бағдарламада **isalpha()**-жолдың әріптерден тұратындығын тексеретін, **isdigit()**- жолдың сандардан тұратындығын тексеретін , **isupper()**-жолдың жоғарғы регистрде таңбалар бар-жоғын тексеретін, **islower()**- кіші регистрде таңбалар бар-жоғын тексеретін әдістер қолданылады.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
print("\n Для создания надежного пароля используйте: \n 1. Заглавные буквы алфавита (от А до Z). \n 2. Строчные буквы алфавита (от а до z). \n 3. Цифры (от 0 до 9). \n 4. Длина пароля >8 символов")
parol=input("\n Введите пароль: ")
c="0123456789"
f1=False
f2=False
f3=False
for i in parol:
    if i.isupper():
        f1=True
    if i.islower():
        f2=True
```

```

    if i in c:
        f3=True
if len(parol)>=8:
    if f1==True:
        if f2==True:
            if f3==True:
                print("Ваш пароль ", parol, " очень надежный")
            else:
                print("Ваш пароль ", parol, " средней сложности, рекомендуем
его сменить")
        else:
            print("Ваш пароль ", parol, " средней сложности, рекомендуем его
сменить")
    else:
        print("Ваш пароль ", parol, " средней сложности, рекомендуем его
сменить")
if 6<len(parol)<8:
    if parol.isalpha()==True:
        if (f1==False) or (f2==False):
            print("Ваш пароль ", parol, " ненадежен, срочно смените его")
    if parol.isdigit()==True:
        print("Ваш пароль ", parol, " ненадежен, срочно смените его")
    if f1==True:
        if f2==True:
            print("Ваш пароль ", parol, " средней сложности, рекомендуем его
сменить")
        else:
            if f3==True:
                print("Ваш пароль ", parol, " средней сложности, рекомендуем
его сменить")
            elif f3==True:
                if f2==True:
                    print("Ваш пароль ", parol, " средней сложности, рекомендуем его
сменить")
if len(parol)<=6:
    print("Ваш пароль ", parol, " ненадежен, срочно смените его")

```

Бағдарламаның нәтижесі 122-ші суретте көрсетілген

.


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\my prog\000008504.py

Для создания надежного пароля используйте:
1. Заглавные буквы алфавита (от А до Z).
2. Строчные буквы алфавита (от а до z).
3. Цифры (от 0 до 9).
4. Длина пароля >8 символов

Введите пароль: Egdskj4465
Ваш пароль Egdskj4465 очень надежный
>>>
RESTART: C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\my prog\000008504.py

Для создания надежного пароля используйте:
1. Заглавные буквы алфавита (от А до Z).
2. Строчные буквы алфавита (от а до z).
3. Цифры (от 0 до 9).
4. Длина пароля >8 символов

Введите пароль: ehdsjke
Ваш пароль ehdsjke ненадежен, срочно смените его
>>>
RESTART: C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\my prog\000008504.py

Для создания надежного пароля используйте:
1. Заглавные буквы алфавита (от А до Z).
2. Строчные буквы алфавита (от а до z).
3. Цифры (от 0 до 9).
4. Длина пароля >8 символов

Введите пароль: eWhhhfa
Ваш пароль eWhhhfa средней сложности, рекомендуем его сменить
>>>
```

Ln: 36 Col: 4

Сурет 122 – Парольдің сенімділігін тексеру бағдарламасының нәтижесі

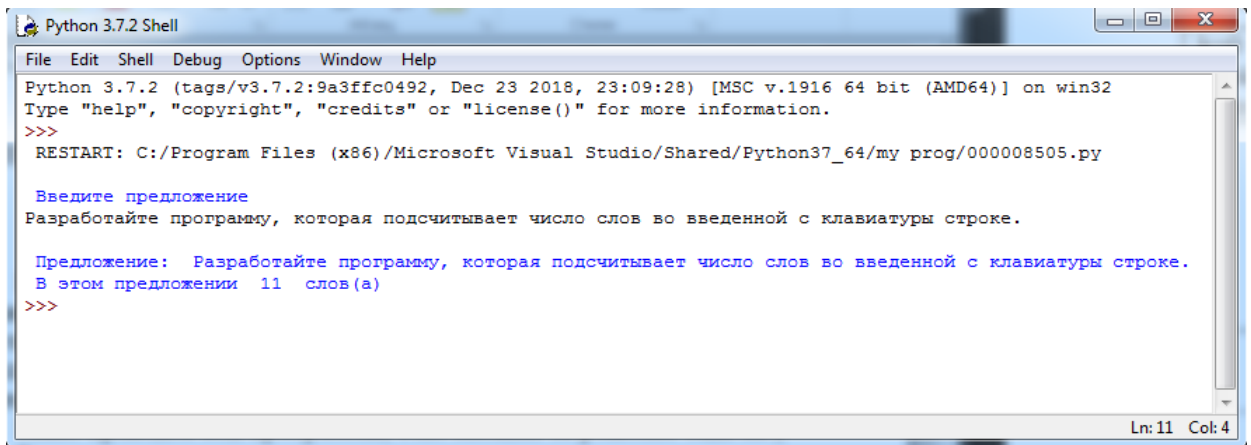
Есеп 8.5.5. Пернетақтадан енгізілген жолдағы сөздердің санын есептейтін бағдарламаны құрыңыз. Біз бұл сөзді басқалардан бос орынмен бөлінген таңбалардың кез-келген реттілігі деп санаймыз.

Шешімі . Capitalize() әдісі жолдың бірінші таңбасын жоғарғы регистрге аударады, сондықтан пайдаланушы жолды кіші әріптермен тере бастаса, нәтиже жолында өзгерістер болады. Белгіш бойынша жолды бөлуді жүзеге асыратын **split (" ")** әдісінің көмегімен (біздің жағдайда - бос орын) енгізілген жолдағы сөздер анықталады.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
k=0
stroka=input("\n Введите предложение \n")
s=stroka.capitalize()
spisok=stroka.split(" ")
for i in range (len(spisok)):
    k=k+1
print("\n Предложение: ", s, "\n В этом предложении ", k, " слов(a)")
```

Бағдарламаның нәтижесі 123-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008505.py

Введите предложение
Разработайте программу, которая подсчитывает число слов во введенной с клавиатуры строке.

Предложение: Разработайте программу, которая подсчитывает число слов во введенной с клавиатуры строке.
В этом предложении 11 слов(a)
>>>
```

Сурет 123 – Мәтіндегі сөздерді санау бағдарламасы жұмысының нәтижесі

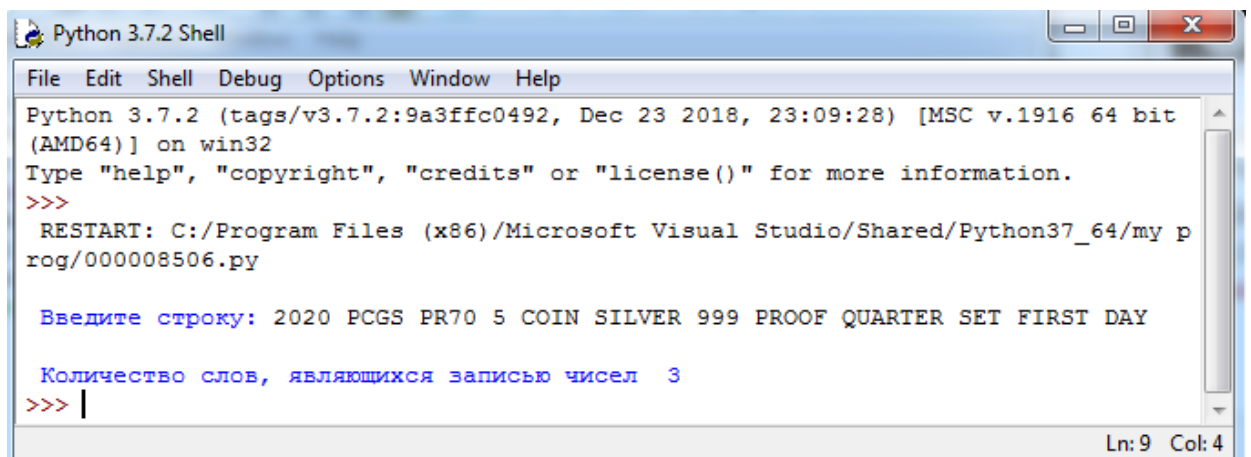
Есеп 8.5.6. Таңбалар жолы берілген. Натурал санның жазбасы болып табылатын сөздердің санын анықтаңыз (тек сандардан тұрады).

Шешімі . Біз `isdigit()` әдісін қолданамыз, оның мақсаты жолдағы сандардың бар-жоғын талдаудан тұрады. **For** операторы бар циклде біз жолдың әр сөзін тексереміз және жағдайдың ақиқаты болған жағдайда есептегішті бірге арттырамыз.

Төменде есепті шешуге жауап беретін бағдарлама коды берілген.

```
k=0
stroka=input("\n Введите строку: ")
spisok=stroka.split(" ")
n=len(spisok)
for i in range (n):
    if spisok[i].isdigit():
        k=k+1
stroka1="" .join(stroka)
print("\n Количество слов, являющихся записью чисел ", k)
```

Бағдарламаның нәтижесі 124-ші суретте көрсетілген



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/my prog/000008506.py

Введите строку: 2020 PCGS PR70 5 COIN SILVER 999 PROOF QUARTER SET FIRST DAY

Количество слов, являющихся записью чисел 3
>>> |
```

Сурет 124 – Мәтіндегі натурал сандарды есептеу бағдарламасының нәтижесі

8.6 Бақылау сұрақтары

1. Әріптік-сандық таңбаларды кодтауды қолдайтын кодтау қалай аталады? Оның құрылымы туралы айтып беріңізші.
2. Таңбалармен жұмыс істеутін негізгі функцияларды атаңыз. Мысалдар келтіріңіз.
3. Жол таңбаларын әртүрлі пернетақта регистрлеріне түрлендіруге мүмкіндік беретін жолдармен жұмыс істеу әдістерін атаңыз.
4. Қандай әдіс жолды ішкі жолдарға бөлуге мүмкіндік береді? Оның синтаксисін жазыңыз.
5. Жолды тізімге түрлендіруге қандай әдіс жауап береді? Оның синтаксисін жазыңыз.
6. Негізгі жол алгоритмдеріне мысалдар келтіріңіз.
7. Жолды қалай кескіндеуге болады?

8.7 Өз бетінше шешуге арналған тапсырмалар

1. Пернетақтадан енгізілген жолдағы сөздер санын есептейтін бағдарлама құрыңыз. Біз басқалардан бос орынмен бөлінген кез-келген таңбалар тізбегін сөз ретінде қарастыруға келісеміз.
2. Пернетақтадан енгізілген таңбалар тізбегі екілік сандар жүйесінде жазылған бүтін сан екенін тексеретін бағдарламаны жасаңыз.
3. Пернетақтадан енгізілген жолдағы сөздердің орташа ұзындығын есептейтін бағдарлама құрыңыз.
4. Пернетақтадан енгізілген мәтінді шифрлайтын бағдарлама құрыңыз. Шифрлау процесі келесідей: он саны пернетақтадан енгізілген әр таңбаның ондық кодынан алынады. Алу нәтижесінде алынған мән компьютер экранында көрсетілетін басқа таңбаның ондық коды ретінде түсіндіріледі.
5. Сөз берілді . Оның палиндром екенін анықтаңыз(екі бағытта да бірдей оқылатын сөз, мысалы, "потоп").
6. Таңбалар жолы берілген. Жолдағы ең ұзын сөзді және осындай ұзындықтағы сөздердің санын анықтаңыз.
7. Таңбалар жолы берілген. Одан барлық бос орындарды алып тастаңыз.
8. Таңбалар жолы берілген. Сөз берілген. Бұл сөзді жолдан алып тастаңыз
9. Таңбалар жолы берілген. Бірінші және екінші нүктелер арасындағы ішкі жолды таңдаңыз.
10. Таңбалар жолы берілген. Ең қысқа және ең ұзын сөздердің ұзындығын анықтаңыз.
11. Таңбалар жолы берілген. Бір әріптен қанша сөз басталып, аяқталатынын анықтаңыз.
12. Таңбалар жолы берілген. Қанша сөзде кем дегенде бір "е" әрпі барын анықтаңыз.
13. Таңбалар жолы берілген. Оның дұрыс жақшаның өрнегі екенін анықтаңыз. Тек дөңгелек жақшаларды қарастырыңыз.

14. Таңбалар жолы берілген. Қанша сөз үш "е" әрпінен тұратынын анықтаңыз.

15. Жолда тек сандар бар. Алда тұрған нөлдерді жою керек.

16. Таңбалар жолы берілген. Жолдағы тыныс белгілерінің санын есептеңіз.

17. Таңбалар жолы берілген. Жолдан барлық үтірлерді алып тастаңыз.

18. Таңбалар жолы берілген. Кейбір сөз берілген. Әр бос орыннан(пробел) кейін сол сөзді қойыңыз.

19. Таңбалар жолы берілген. Жолда кездесетін сандардың қосындысын табыңыз.

20. Таңбалар жолы берілген. Жолдан барлық сандарды жойыңыз.

21. Таңбалар жолы берілген. Жолдан ең ұзын сөзді алып тастаңыз.

22. Таңбалар жолы берілген. Әр тыныс белгісінен кейін жолға бос орын қойыңыз.

23. Сандар мен латын әріптерінен жол берілген. Осы жолда қай әріптердің - дауысты дыбыстардың (А,Е,І,О және т.б.) немесе дауыссыз дыбыстардың көбірек екенін анықтаңыз.

24. Берілген жолдан оның тырнақшада орналасқан бөліктерін алып тастаңыз (тырнақшалармен бірге).

25. Сөз берілген. Бұл сөзде А әріпін О әріпімен ауыстырып. Егер бұл сөзде А әрпі болмаса, онда тиісті хабарламаны шығарыңыз.