Дәріс №7. Python. Жиын Жолмен, файлмен жұмыс

Жол дегеніміз – ұзындығы 255-тен артпайтын символдар тізбегі. Тізбектің мағынасы болуы міндет емес. Мысал: ‘df56’, ‘*d-шар56‘, ‘*-4 лд’

Жолдар *string* жолдық типпен сипатталады. Жолдық типті анықтағанда ондағы символдар санын көрсетуге болады.

Жазылу форматы:

type

<типтің аты>=string [символдар саны];

var <идентификатор>: <типтің аты>;

String типтегі айнымалыны типті алдын-ала сипаттамай-ақ көрсетуге болады: **var** <идентификатор>:string[жолдың ұзындығы];

N символдан тұратын жолға жадыдан N+1 байт бөлінеді. N байт-символдарды сақтау үшін, ал бір байт – жолдың ұзындығын сақтау үшін.

Жолдық өрнектер. Олар жолдық тұрақтылардан, айнымалылардан, функциялардан және операция таңбаларынан тұрады. Мысал:’ма’+’ма’

Жолдық процедуралар мен функциялар

Аты, жазылуы	Қызметі	Ескерту
1. **жою** delete(a,p,n)	a жолындағы p пози-циядан бастап, ұзындығы n символды жояды.	p<=255; нәтиже басқа айны-малыға меншіктелмейді.
2. **кіргізу** insert (a,s,p)	a жолын s жолына p по-зициядан бастап кіргізеді.	нәтиже басқа айны-малыға меншіктелмейді.
3. **типті ауыстыру** str(x,a)	x сандық шаманы жолға өңдеп, a-ға меншіктейді	x-ті шығару форматымен жазуға болады. X сандық типте, a –жолдық типте.
4. **типті ауыстыру** val(a,x,c)	a жолын сандық шамаға өңдеп, x айнымалыға орналастырады. A жо-лында бос символ болмау керек.	c-өңдеу нәтижесі, бүтін сан. Егер өңдеуде қате болмаса c=0 болады. A:string, x сандық типте, c:integer.
5.**ұзындық** length(a)	a жолының ұзындығын табады.	Нәтижені айнымалыға меншіктеуге болады.
6. **ретімен тіркестіру** concat(a,b,..s)	a,b,..s жолдарын сол ретімен тіркестіреді	-
7. **белгілеу** copy (a,p,n)	a жолынан p позициядан бастап, ұзындығы n символды белгілейді.	егер p>length(a) болса, нәтижесі бос символ; p>255 болса, қате.
8. **позиция** pos (a,s)	a жолы s жолында нешінші позицияда тұрғанын табады.	егер a жолы s жолында болмаса, нәтиже=0
9.**регистрді ауыстыру** upcase(ch)	кіші әріпті бас әріпке өзгертеді.	тек латын алфавитін ғана.

1. Цезарь әдісі. Мысалда = -1 кілтімен «криптография, бұл ғылым» сөзінің кодтау нәтижесі көрсетілген. Декодтау бағдарламасын жазу керек (оқытушы декодтау сөзін береді; шифрланған кілтті, сөзді анықтау).

```
Вариант 23: СИБ-15-1, студент Иванов И.И.
----------------------------------------
Это программа КОДИРОВАНИЯ методом Цезаря
введите текст для шифрования:криптография - это наука
вы ввели: криптография - это наука   ключ шифрования= -1
посимвольная разбивка и расшифровка:
к = 1082
р = 1088
и = 1080
п = 1087
т = 1090
о = 1086
г = 1075
р = 1088
а = 1072
ф = 1092
и = 1080
я = 1103
  = 32
- = 45
  = 32
э = 1101
т = 1090
о = 1086
  = 32
н = 1085
а = 1072
у = 1091
к = 1082
а = 1072
После наложения ключа, ответ:
йпзоснвпяузю , ьсн мятйя
```

2. Полибиан шаршысы (тікбұрыш).

Өлшемі 7*5 абстрактты матрицада бағдарлама жұмысының листингі және нәтижесі көрсетілген; студентке тапсырманы орындау барысында өзіне ыңғайлы ету және кестенің басына өзінің тегі және атындағы әріптерді қайталаусыз жазу керек, 1 кестеде көрсетліген сияқты өзгерту керек. Одан кейін матрицадан кодтау кезінде алынған барлық цифрларды жақшасыз және «,» таңбасынсыз жазу керек. Одан кейін барлық сандарды бір қатарға
– мысалы түрінде жалғастыру нәтижесі көрсетілген. Бұл Полибий шаршысының көмегімен кодталған сөз болып табылады.

1 кесте – Полибий шаршысының көмегімен кодтау мысалы

	1	**2**	**3**	**4**	**5**	**6**
1	З	У	Е	В	А	К
2	Т	Р	И	Н	Б	Г
3	Д	Ж	Л	М	О	П
4	Р	С	Т	Ф	Х	Ц
5	Ч	Ш	Щ	Ъ	Ь	Э
6	Ю	Я	-	,	.	_

а)

```
matrix = [['а','б','в','г','д'],
          ['е','ж','з','и','к'],
          ['л','м','н','о','п'],
          ['р','с','т','у','ф'],
          ['х','ц','ч','ш','щ'],
          ['ъ','ы','ь','э','ю'],
          ['я','-',',','.',' ']]
def polibii(bukva):
    for i in range(len(matrix)):
        for j in  range(len(matrix[i])):
            if bukva ==  matrix[i][j]:
                return i+1, j+1
print("Вариант 23: СИБ-15-1, студент Иванов И.И.
print("----------------------------------------")
print("Это программа КОДИРОВАНИЯ Полибианским квадратом")
text=input("введите текст для шифрования:")
print("вы ввели фразу:",text)
for bukva in text.lower().replace('j','i'):
    print (polibii(bukva),end='')
```

```
Вариант 23: СИБ-15-1, студент Иванов И.И.
----------------------------------------
Это программа КОДИРОВАНИЯ Полибианским квад
ратом
введите текст для шифрования:абвгдежзиклм
вы ввели фразу: абвгдежзиклм
(1, 1)(1, 2)(1, 3)(1, 4)(1, 5)(2, 1)(2, 2)(
2, 3)(2, 4)(2, 5)(3, 1)(3, 2)
```

```
import re
var = u'abce4387def..//-+,,,,zzqw5>?'
print("первоначальная строка:", var)
result=re.sub(u'[^a-z\s]*', u'', var)
print("полученная строка:", result)
```

```
>>>
первоначальная строка: abce4387def..//-+,,,,zzqw5>
полученная строка: abcedefzzqw
```

Декодтау. Қандай да бір сандар қатары беріледі және мәндер матрицасын біле отырып (1 кестеге ұқсас) қандай сөз жасырылғандығын анықтау керек. Барлық сандар бір цифрдан

тұратындығын ескере отырып, декодтау кері процедурасы (жұп бойынша бөлу бір мәнді болады).

```
Вариант 23: СИБ-15-1, студент Иванов И.И.
----------------------------------------
Программа ДЕКОДИРОВАНИЯ Полибианским квадратом
введите строку для дешифрования:3511411115342542
вы ввели цифры: 3511411115342542
разбивка по парам: 3,5 1,1 4,1 1,1 1,5 3,4 2,5 4,2
ОТВЕТ - задуманная фраза: парадокс
```

3. Вижинер шифры.

Кілт ретінде студенттің тегі және аты (әріптерін қайталамай) болып келеді; сөйлемге кілт қойылады, егер кілттің ұзындығы кішкентай болса, онда кілт көшіріледі.

```
print("Вариант 23: СИБ-15-1, студент Иванов И.И."); print(42*"-")
print("Это программа КОДИРОВАНИЯ методом Вижинера")
a=input("введите фразу для шифрования:"); n=len(a); key=('зуевактрин')
m=len(key); d=n//m; e=n%m; print("первоначальный ключ:",key)
print("ключ накладывается",d,"раз и",e,"букв")
if e==1:
    x=key[0];
    c=d*key+x
elif e==2:
    x=key[0]+key[1]
    c=d*key+x
elif e==3:
    x=key[0]+key[1]+key[2]
    c=d*key+x
elif e==4:
    x=key[0]+key[1]+key[2]+key[3]
    c=d*key+x
elif e==5:
    x=key[0]+key[1]+key[2]+key[3]+key[4]
    c=d*key+x
elif e==6:
    x=key[0]+key[1]+key[2]+key[3]+key[4]+key[5]
    c=d*key+x
elif e==7:
    x=key[0]+key[1]+key[2]+key[3]+key[4]+key[5]+key[6]
    c=d*key+x
elif e==8:
    x=key[0]+key[1]+key[2]+key[3]+key[4]+key[5]+key[6]+key[7]
    c=d*key+x
elif e==9:
    x=key[0]+key[1]+key[2]+key[3]+key[4]+key[5]+key[6]+key[7]+key[8]
    c=d*key+x
elif e==10:
    x=key[0]+key[1]+key[2]+key[3]+key[4]+key[5]+key[6]+key[7]+key[8]+key[9]
    c=d*key+x
else:
    print(c)
print("и полученный ключ =",c); print("ОТВЕТ после наложения ключа с текстом:")
for i in range (n):
    f=ord(a[i])+ord(c[i])
    if f>1103:
        q=f-1103; w=chr(q); w.lower(); print(w.lower(),end='')
```

```
Вариант 23: СИБ-15-1, студент Иванов И.И.
----------------------------------------
Это программа КОДИРОВАНИЯ методом Вижинера
введите фразу для шифрования:пример фразы шифрования
первоначальный ключ: зуевактрин
ключ накладывается 2 раз и 3 букв
и полученный ключ = зуевактринзуевактринзуе
ОТВЕТ после наложения ключа с текстом:
чдопжы‼ещопп–ыйягялохье
```

Декодтау кезінде шифрланған фразаны және бірінші кілтті біле отырып, алғшқы мәтінді қалпына келтіру керек.

```
Вариант 23: СИБ-15-1, студент Иванов И.И.
----------------------------------------
Это программа ДЕКОДИРОВАНИЯ методом Вижинера.
введите фразу для дешифрования (полученную наложением ключа и текста):ихижершшсшуаусрыдгь
первоначальный ключ ( зуевактрин ) накладывался 1 раз и 9 букв
а значит наложен ключ был = зуевактринзуевактри
ОТВЕТ после отделения ключа от текста:
абвгдежзиклмнопрсту
```