

ПРАКТИЧНА РОБОТА №3. СТРУКТУРИ ДАНИХ

Мета роботи – вивчити та засвоїти базові навички використання базових структур даних у Python. Дізнатися про принципи взаємодії з рядками та списками, кортежами та словниками, а також про основні вбудовані методи для їх обробки.

Основні теоретичні відомості

Рядки

Базові поняття про рядки

Рядок – це послідовність символів з довільним доступом. Рядки в мові Python неможливо змінити - в цьому випадку говорять, що це immutable тип. Спроба змінити символ в певній позиції або підстроку викличе помилку:

```
word = 'GG EZ'
word[1] = 'J'
```

```
TypeError                                 Traceback (most recent call last)
<ipython-input-1-0c6fa3432d82> in <module>
      1 word = 'GG EZ'
----> 2 word[1] = 'J'
```

TypeError: 'str' object does not support item assignment

Рисунок 3.1 – Демонстрація спроби зміни рядка

Але якщо потрібно, то змінити можна, наприклад, так:

```
[6]: word = 'GG EZ'
      word = word[:2] + '!' + word[3:]
      print(word)
```

GG!EZ

Рисунок 3.2 – Демонстрація зміни рядка

Індекси можуть мати негативні значення для відліку з кінця - відлік починається з -1:

```
[7]: print(word[-1])
```

z

Рисунок 3.3 – Отримання значення рядка за індексом

Рядки в Python можна укладати як в одинарні, так і в подвійні лапки, причому лапки одного типу можуть бути довільно вкладені в лапки іншого типу:

```
[9]: word1 = '123'  
word2 = "7'8''9"  
print(word1)  
print(word2)
```

123
7'8''9

Рисунок 3.4 – Укладання рядків в лапки

Великі набори рядків і цілі тексти можна укладати в потрібні лапки:

```
[11]: print("""  
One  
Two  
Three  
""")
```

One
Two
Three

Рисунок 3.5 – Укладання рядка в потрібні лапки

Зворотний слеш в рядках використовується для так званої escape-послідовності. Після слеша може іти один або кілька символів. У наступному прикладі комбінація '\n' - це новий рядок, '\t' - це табуляція:

```
[12]: s = 'HI\nGL\tHF'
      print(s)

HI
GL      HF
```

Рисунок 3.6 – Приклад використання escape-послідовності

Зрізи

Зріз – це механізм гнучкого управління рядком на основі індексації. Можна отримати будь-який символ рядка по його індексу. Перший символ має індекс 0. Підрядок може бути визначена за допомогою зрізу - двох індексів, розділених двокрапкою:

```
[15]: word3 = 'Hello people'
      print(word3[4])
      print(word3[0:4])
      print(word3[4:8])

o
Hell
o pe
```

Рисунок 3.7 – Приклад використання зрізів

Якщо в зрізі опущений перший символ, значить, він дорівнює нулю; якщо опущений останній символ - він дорівнює довжині рядка:

```
[17]: print(word3[:5])
      print(word3[6:])

Hello
people
```

Рисунок 3.8 – Приклад використання зрізів

Можна вибирати послідовність символів з рядка з певним кроком:

```
[18]: print(word3[::2])
      print(word3[::-1])

Hlopol
elpoep olleH
```

Рисунок 3.9 – Приклад вибору послідовності з певним кроком

Операції з рядками

Рядки можна склеювати за допомогою оператора "+" (між двох рядків поспіль взагалі можна нічого не ставити, і вони будуть сконкатеновані), рядки можна множити за допомогою оператора "*", а також рядки можна порівнювати за допомогою операторів <, <=, ==, !=, >, >=.

```
[22]: word4="Hello" + "Word"
      word5= "Hello " "People"
      print(word4)
      print(word5)
      word5*=3
      print(word5)
      if(word5==word4):
          print("Yes")
      else:
          print("No")

HelloWord
Hello People
Hello PeopleHello PeopleHello People
No
```

Рисунок 3.10 – Використання операцій з рядками

Методи для обробки рядків

Є велика кількість методів для обробки рядків, далі будуть наведені основні з них:

find – знаходить підрядок в рядку та повертає позицію входження рядка, або -1:

```
word6='I have no friends'
buf=(word6.find('friends'))
print(buf)

10
```

Рисунок 3.11 – Приклад використання find

join – об'єднує через роздільник набір рядків:

```
6]: word7=['I', 'friends']
    sep1 = '+'
    sep2 = '='
    word9=(sep1.join(word7))
    print(word9)
    word8=[word9, 'clever friends']
    word9=(sep2.join(word8))
    print(word9)

I+friends
I+friends=clever friends
```

Рисунок 3.12 – Приклад використання join

split – це зворотна функція для **join**, розбиває рядок на послідовність:

```
print(word9.split('+'))

['I', 'friends=clever friends']
```

Рисунок 3.13 – Приклад використання split

replace – замінює в рядку одну підстроку на іншу:

```
word10=word9.replace('I', ' another')
print(word10)

another+friends=clever friends
```

Рисунок 3.14 – Приклад використання replace

strip – видаляє пробіли зліва і справа:

```
print("( "+word7[0]+" ")")
print("(" +word7[0].strip(" ")+"")")

( I )
(I)
```

Рисунок 3.15 – Приклад використання strip

Форматування рядків

Для форматування в рядках використовується стандартний оператор – символ «%». Зліва від відсотка вказуємо рядок, праворуч – значення або список значень:

```
[55]: s = 'Hello %s' % 'word'
      print(s)
      s = 'one %d %f' % (2, 3.5)
      print(s)
      x = 4/3
      print('%10.5f' % x)
      '%10.5f' % x

Hello word
one  2 3.500000
      1.33333
[55]: '  1.33333'
```

Рисунок 3.16 – Приклад форматування рядків

Списки

Базові поняття про списки

Для угруповання безлічі елементів в Python використовується список (list), який може бути записаний як індексована послідовність значень, розділених комами, укладена в квадратні дужки. Списки мають довільну вкладеність, тобто можуть включати в себе будь-які вкладені списки. Фізично список являє собою масив покажчиків (адрес) на його елементи.

Елементи списку не обов'язково повинні бути одного типу.

```
[62]: mas = ['hello', 'word', 65, 12.34, 5]
      print(mas)

['hello', 'word', 65, 12.34, 5]
```

Рисунок 3.17 – Створення списку

Як і для рядків, для списків нумерація індексів починається з нуля. Для списку можна отримати зріз, об'єднати кілька списків і так далі:

```
[64]: print(mas[3])
      print(mas[0:2])
      print(mas[1:3])

12.34
['hello', 'word']
['word', 65]
```

Рисунок 3.18 – Використання зрізів на списках

Можна міняти як окремі елементи списку, так і діапазон:

```
[66]: mas[1]="people"
      mas[2:5]=[2,4]
      print(mas)

['hello', 'people', 2, 4]
```

Рисунок 3.19 – Зміна елементів списку

Також можна робити вставку у довільне місце у списку:

```
[79]: mas[1:1]=["funny"]
      print(mas)

['hello', 'funny', 'people', 2, 4]
```

Рисунок 3.20 – Вставлення у довільне місце у списку

Операції зі списками

```
[91]: mas1=mas[:]      #створення копії списку
      mas2=list(mas)   #створення копії списку
      mas3=mas         #створення копії посилання на список
      mas[1]=1
      print(mas)
      print(mas1)
      print(mas2)
      print(mas3)

['hello', 1, 'people', 2, 4]
['hello', 'funny', 'people', 2, 4]
['hello', 'funny', 'people', 2, 4]
['hello', 1, 'people', 2, 4]
```

Рисунок 3.21 – Створення копії списку

```
[92]: mas4=mas3+mas2  
      print(mas4)  
      ['hello', 1, 'people', 2, 4, 'hello', 'funny', 'people', 2, 4]
```

Рисунок 3.22 – Конкатенація списків

```
[93]: mas5=mas2*3  
      print(mas5)  
      ['hello', 'funny', 'people', 2, 4, 'hello', 'funny', 'people', 2, 4, 'hello', 'funny', 'people', 2, 4]
```

Рисунок 3.23 – Множення або повторення списків

Вбудовані функції

Списки мають великий набір функцій:

- **append, extend** - додавання;
- **insert** - вставка;
- **index** - знайти індекс першого входження конкретного елемента;
- **count** - підрахунок повторів елемента;
- **remove, del** - видалення елемента;
- **sort** - сортування;
- **reverse** - реверс;
- **pop** - витяг елемента;
- **len** - довжина списку;
- **max** - максимальний елемент;
- **min** - мінімальний елемент;
- **in** - перевірка елемента на входження.

Додавати можна як одинарні елементи, так і набір елементів. Списки можуть бути вкладеними – вкладений список.


```
[9]: mas6 = ['hi', 'I', "am", 'here']
mas.append(mas6)
print(mas)

['hello', 1, 'people', 2, 4, ['hi', 'I', 'am', 'here']]

[20]: mas.insert(5, 'chh')
print(mas)

['hello', 1, 'people', 2, 4, 'chh', ['hi', 'I', 'am', 'here']]

[40]: print(mas.count(1))

2
```

Рисунок 3.24 – Демонстрація виконання функцій

```
[37]: print(mas.index('people'))

2

[36]: if "people" in mas:
      print("true")
      else:
        print("false")
      if "cat" in mas:
        print("true")
      else:
        print("false")

true
false

[41]: mas.remove('people')
print(mas)

['hello', 1, 2, 4, 1, ['hi', 'I', 'am', 'here']]

[42]: del mas[5][3]
print(mas)

['hello', 1, 2, 4, 1, ['hi', 'I', 'am']]

[49]: mas_int=[1,3,5,4,7,2,1,5,6,8,4,2,4]
mas_int.sort()
print(mas_int)

[1, 1, 2, 2, 3, 4, 4, 4, 5, 5, 6, 7, 8]

[51]: mas_int.reverse()
print(mas_int)

[8, 7, 6, 5, 5, 4, 4, 4, 3, 2, 2, 1, 1]

[52]: print(len(mas_int))
print(max(mas_int))
print(min(mas_int))

13
8
1

[54]: print(mas_int.pop())
print(mas_int)
print(mas_int.pop(0))
print(mas_int)

1
[8, 7, 6, 5, 5, 4, 4, 4, 3, 2, 2, 1]
8
[7, 6, 5, 5, 4, 4, 4, 3, 2, 2, 1]
```

Рисунок 3.25 – Демонстрація виконання функцій

Кортеж

Базові поняття про кортежі

Список так само може бути незмінним (immutable), як і рядок, в цьому випадку він називається кортеж (tuple). Кортеж використовує менше пам'яті, ніж список. Кортеж замість квадратних дужок використовує круглі (хоча можна і зовсім без дужок). Кортеж не допускає змін, в нього не можна додати новий елемент, хоча він може містити об'єкти, які можна змінити:

```
[4]: t = 1,[2,3]
      print(t)
      t[1] = 2

(1, [2, 3])

-----
TypeError                                 Traceback (most recent call last)
<ipython-input-4-aeb14024b19c> in <module>
      1 t = 1,[2,3]
      2 print(t)
----> 3 t[1] = 2

TypeError: 'tuple' object does not support item assignment

[3]: t[1].append(4)
      print(t)

(1, [2, 3, 4, 4])
```

Рисунок 3.26 – Зміна елементу кортежу

Функція `tuple()` бере в якості аргументу рядок або список і перетворює його в кортеж:

```
[5]: t=tuple('abc')
      print(t)

('a', 'b', 'c')
```

Рисунок 3.27 – Перетворення рядка у кортеж

До можливих операцій над кортежами відносяться всі операції над списками, що не змінюють список (додавання, множення на число, методи

index() і count() і деякі інші операції). Можна також по-різному змінювати елементи місцями і так далі.

Словник

Базові поняття про словники

Після списків словник є самим гнучким вбудованим типом. Якщо список - це впорядкована колекція, то словник - невпорядкована. Основні особливості словників:

- Доступ здійснюється по ключу, а не за індексом. За аналогією зі списком, в словнику можна отримати доступ до елементів в циклі по ключам.
- Значення словника зберігаються в невідсортованому порядку, більш того, ключі можуть зберігатися не в тому порядку, в якому вони додаються.
- За аналогією зі списками, словник може зберігати вкладені словники. Словник може зберігати в якості значень об'єкти будь-якого типу (heterogeneous). Ключ в словнику - immutable тип, може бути рядком, цілим числом, float або кортежем, що складається із зазначених типів.
- Словники реалізовані як хеш-таблиці з швидким доступом.
- Словники, так само як і списки, зберігають посилання на об'єкти, а не самі об'єкти.

Словник (dictionary) – це асоціативний масив або хеш. Це неврегульована безліч пар ключ: значення з вимогою унікальності ключів. Пара фігурних дужок {} створює порожній словник. На відміну від послідовностей, доступ до елементів словника проводиться по ключу, а не за індексом, ключ може бути будь-якого типу, ключ не допускає змін.

Основні операції над словником - збереження з заданим ключем і витяг з нього значення. Також можна видалити пару key: value за допомогою інструкції del.

Метод keys() для словника повертає список всіх використовуваних ключів в довільному порядку; для сортування списку потрібно застосувати метод sort(). Для визначення наявності певного ключа є метод has_key(), який у версії 3.0 встигне застаріти - замість нього є оператор in. Додавання нового об'єкта в

словник не вимагає попередніх перевірок: якщо раніше ключу вже відповідало деяке значення, воно буде перезаписано.

```
[10]: tel = {'Lola' : 380950505095, 'BuBochKA' : 380503223323}
      tel['Lapusya'] = 380990767676
      print(tel)
      print(tel['Lapusya'])
      del tel['Lola']
      print(tel)
      print(tel.keys())

{'Lola': 380950505095, 'BuBochKA': 380503223323, 'Lapusya': 380990767676}
380990767676
{'BuBochKA': 380503223323, 'Lapusya': 380990767676}
dict_keys(['BuBochKA', 'Lapusya'])
```

Рисунок 3.28 – Приклад створення словника

Створити словник можна декількома способами:

Звичайний вираз - це зручно, якщо словник статичний:

```
[12]: T = {'name': 'Alexandro', 'age': 35}
      print(T)

{'name': 'Alexandro', 'age': 35}
```

Рисунок 3.29 – Перший варіант створення словника

Динамічний варіант створення на льоту:

```
[13]: D = {}
      D['name'] = 'Alexandro'
      D['age'] = 35
      print(D)

{'name': 'Alexandro', 'age': 35}
```

Рисунок 3.30 – Перший варіант створення словника

За допомогою функції `dict()` - ключі при цьому повинні бути рядками. За допомогою цієї функції можна позбавити себе від обов'язкової умови укладати ключ в лапки.

```
[17]: d1 = dict(id=1948, name="Washer", size=3)
      d2 = dict({"id": 1948, "name": "Washer", "size": 3})
      d3 = dict([("id", 1948), ("name", "Washer"), ("size", 3)])
      d4 = dict(zip(("id", "name", "size"), (1948, "Washer", 3)))
      print(d1,d2,d3,d4, sep='\n')
```

Рисунок 3.31 – Створення словника за допомогою функції `dict`

За допомогою `fromkeys ()` - створює словник за списком ключів з порожніми значеннями:

```
[18]: D = {}.fromkeys(['name', 'age'],123)
      print(D)

{'name': 123, 'age': 123}
```

Рисунок 3.32 – Створення словника за допомогою функції `fromkeys`

За допомогою конструктора:

```
[21]: d = dict((x, x**2) for x in range(5))
      print(d)

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

Рисунок 3.33 – Створення словника за допомогою конструктора

Функції та методи словників

- **`dict ()`** - створення словника;
- **`len ()`** - повертає число пар;
- **`clear ()`** - видаляє всі значення зі словника;
- **`copy ()`** - створює псевдокопію словника;
- **`deepcopy ()`** - створює повну копію словника;

- **fromkeys ()** - створення словника;
- **get ()** - отримати значення по ключу;
- **has_key ()** - перевірка значення по ключу;
- **items ()** - повертає список значень;
- **iteriyems ()** - повертає ітератор;
- **keys ()** - повертає список ключів;
- **iterkeys ()** - повертає ітератор ключів;
- **pop ()** - витягує значення по ключу;
- **popitem ()** - витягує довільне значення;
- **update ()** - змінює словник;
- **values ()** - повертає список значень;
- **itervalues ()** - повертає ітератор на список значень.
- **in** - оператор, перевіряє наявність значення по ключу;
- **del** - оператор, видаляє пару по ключу;
- **dict ()** - конструює словник за допомогою послідовності.

Оскільки словники є мапи (map), а не послідовності, до них не можна застосувати конкатенацію або зрізи. До словників можна застосовувати стандартні оператори порівняння: <, <=, ==, !=, >=, > .

Для того щоб зробити прохід по ключам словника, використовуємо for:

```
[22]: table = {'Python': 'Guido van Rossum',
              'Perl': 'Larry Wall',
              'Tcl': 'John Ousterhout' }
for lang in table:
    print(lang, table[lang])

Python Guido van Rossum
Perl Larry Wall
Tcl John Ousterhout
```

Рисунок 3.34 – Перебір елементів з циклом for

За допомогою словників можна зберігати структуровану інформацію у вигляді записів:

```
[23]: man = {'name': 'Serg',  
            'jobs': ['programmer', 'writer'],  
            'web': 'www.iakovlev.org',  
            'home': {'city': 'Moscow', 'zip': 129000}}  
print(man['name'])  
print(man['jobs'][1])  
  
Serg  
writer
```

Рисунок 3.35 – Збереження інформації у вигляді записів

Завдання до виконання

- 1) Виконати завдання, пов'язані із рядками (табл. 3.1);
- 2) Використовуючи словники реалізувати довідник з тематикою відповідно до варіанту. Для довідника реалізувати функції:
 - Додавання;
 - Видалення;
 - Редагування;
 - Демонстрації всього довідника.

А також розрахункову функцію відповідно до варіанту (табл. 3.2).

Таблиця 3.1 – Варіанти завдань

№	Завдання
1	Дано рядок A=" if two witches were watching two watches, which witch would watch which watch?" – замінити всі голосні літери цього рядка порядковим номером цієї літери в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задані дві матриці однакового розміру $X=(5 \times 5)$, $Y=(5 \times 5)$ зі значеннями елементів від -10 до 10. Сформувати матрицю Z, що є сумою матриць X та Y.
2	Дано рядок A=" the thirty-three thieves thought that they thrilled the throne throughout thursday. " – замінити всі перші літери кожного слова цього рядка порядковим номером цієї літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задані дві матриці однакового розміру $X=(4 \times 5)$, $Y=(5 \times 4)$ зі значеннями елементів від -10 до 10. Сформувати матрицю Z, що є добутком матриць X та Y.
3	Дано рядок A=" You know New York, you need New York, you know you need unique New York. " – замінити всі літери верхнього регістру порядковим номером наступного символу в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задані дві матриці однакового розміру $X=(6 \times 3)$, $Y=(6 \times 3)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(6 \times 3)$, яка буде формуватися перенесенням відповідного стовпчику з матриць (X або Y) в якому середнє арифметичне цього стовпця буде більшим (при однакових значеннях віддавати перевагу матриці X)
4	Дано рядок A=" he thrusts his fists against the posts and still insists he sees the ghosts." – замінити всі входження підрядка "th" на "st" і навпаки, замінити кожен пробіл на номер наступної за ним літери в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задані дві матриці однакового розміру $X=(5 \times 3)$, $Y=(5 \times 3)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(5 \times 3)$, яка буде формуватися перенесенням відповідного елементу

№	Завдання
	з матриць (X або Y) модуль якого буде більшим(при однакових значеннях віддавати перевагу Y).
5	Дано рядок A="i wish to wish the wish you wish to wish, but if you wish the wish the witch wishes, i won't wish the wish you wish to wish. " – замінити всі входження підрядка "wi" на "sh" і навпаки, замінити кожен пробіл на номер наступної за ним літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задані дві матриці однакового розміру $X=(4 \times 4)$, $Y=(4 \times 4)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(2 \times 4)$, першим рядком якої буде список середніх арифметичних рядків матриці X, а другий список середніх арифметичних стовпців матриці X
	Дано рядок A="can you can a canned can into an uncanned can like a canner can can a canned can into an uncanned can?" – підрахувати у рядку усі подвоєння літер, також вивести усі варіанти подвоєнь які зустрічаються у рядку, замінити усі подвоєння на порядковий номер цієї літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задані дві матриці однакового розміру $X=(3 \times 4)$, $Y=(3 \times 4)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(3 \times 4)$, яка буде формуватися перенесенням відповідного рядку з матриці (X або Y) в якому середнє арифметичне цього рядка буде меншим (при однакових значеннях віддавати перевагу матриці Y)
6	Дано рядок A="if you must cross a course cross cow across a crowded cow crossing, cross the cross coarse cow across the crowded cow crossing carefully." – замінити в кожному слові рядка, всі букви які зустрічаються більше одного разу (в межах слова) на їх номер в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Підрахувати суму всіх отриманих в рядку чисел.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(6 \times 6)$, виконати дзеркальне відображення матриці відносно головної діагоналі.
	Дано рядок A="brisk brave brigadiers brandished broad bright blades, blunderbusses, and bludgeons balancing them badly. " – замінити всі входження підрядка "br" на "bl" і навпаки, замінити кожен пробіл на номер наступної за ним літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
7	Задані дві матриці однакового розміру $X=(4 \times 5)$, $Y=(4 \times 5)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(4 \times 5)$, кожен елемент якої буде формуватися як сума модулів відповідних елементів із матриць, зі знаком «+», якщо знаки елементів були однаковими, та «-» у зворотному випадку.
	Дано рядок A="imagine an imaginary menagerie manager managing an imaginary menagerie. " – в кожному слові рядка розташувати слова в зворотному порядку, пробіли між словами замінити на порядковий номер пробілу в реченні.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
8	Задана матриця $Z=(5 \times 5)$, помножити кожен елемент на модуль середнього арифметичного рядку в якому він розташований.

№	Завдання
9	Дано рядок A="six sick hicks nick six slick bricks with picks and sticks." – замінити всі слова коротше чотирьох символів на номер входження слова в речення.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(7 \times 7)$, виконати дзеркальне відображення матриці відносно центрально рядка.
10	Дано рядок A="send toast to ten tense stout saints' ten tall tents." – замінити кожну останню букву слова на літеру, а кожну першу на її номер в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(4 \times 6)$, помножити кожен елемент на модуль середнього арифметичного стовпчика в якому він розташований.
11	Дано рядок A="how can a clam cram in a clean cream can?" – замінити кожне друге слово речення на порядковий номер слова в реченні.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(4 \times 5)$, виконати зрушення кожного стовпця на 2 позиції вниз.
12	Дано рядок A="i wish to wash my irish wristwatch." – замінити всі голосні літери цього рядка порядковим номером цієї літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(5 \times 5)$, виконати дзеркальне відображення матриці відносно центрально стовпчика.
13	Дано рядок A="how can a clam cram in a clean cream can?" – замінити всі останні літери кожного слова цього рядка порядковим номером цієї літери в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(4 \times 5)$, виконати зрушення кожного рядку на 2 позиції вліво.
14	Дано рядок A="send toast to ten tense stout saints' ten tall tents." – замінити центральну букву кожного слова на її порядковий номер в алфавіті B="zyxwvutsrqponmlkjihgfedcba" (якщо в слові парна кількість букв - замінити дві центральні).
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(6 \times 6)$, виконати дзеркальне відображення матриці відносно побічної діагоналі.
15	Дано рядок A="six sick hicks nick six slick bricks with picks and sticks." – замінити кожну першу літеру слова на літеру верхнього регістру, а кожну останню на її номер в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Y=(5 \times 5)$, створити вектор Z, що складається з максимальних елементів кожного стовпця матриці.

Таблиця 3.2 – Варіанти завдань

№	Завдання
1	Довідник - « Аеропорт » Поля - [ПШБ] [Рейс] [Клас] [Місце] [Вартість квитка]
	Вивести білети з вартістю нижче ніж середня вартість квитка
2	Довідник - « Пошта » Поля - [ID відправлення] [Відправник] [Одержувач] [Адреса] [Вага]
	Вивести відправлення з вагою більшою за N (N вводити з клавіатури)
3	Довідник - « Квітковий магазин » Поля - [Назва квітки] [Кількість на складі] [Вартість за шт.] [Дата поставки] [Термін зберігання]
	Вивести усі квіти залишок яких на складі менший за N (N вводити з клавіатури)
4	Довідник - « Транспортні компанії » Поля - [Назва] [Кількість Авто] [Вартість 1км перевезення] [Адреса] [Макс. допустима вага]
	Вивести компанії яких Макс. допустима вага більша за N (N вводити з клавіатури)
5	Довідник - « Футбольний Матч » Поля - [Команда 1] [Команда 2] [Рахунок] [Попереджень] [Видаленнь]
	Вивести матчі у яких суддя діставав картку більше ніж N разів (N вводити з клавіатури)
6	Довідник - « Магазин Техніки » Поля - [ID товару] [Назва] [Вартість за шт.] [Кількість на складі] [Термін гарантії (місяців, або років)]
	Вивести усі товари гарантія на які більша за N (N вводити з клавіатури)
7	Довідник - « Beauty bloggers » Поля - [Нікнейм] [Назва каналу] [Посилання] [Вік] [Кількість підписчиків]
	Вивести N найпопулярніших блогерів за зростанням віку (N вводити з клавіатури)
8	Довідник - « Кінолог » Поля - [Порода] [Середня вага] [Середній мак. Вік] [Регіон розповсюдження] [Середня вартість]
	Вивести усі породи у яких середня вага менше за N, а середній вік більший за M (M-N ввести з клавіатури)
9	Довідник - « Бабусині заготовки » Поля - [Назва] [Об'єм] [Рік] [Вид] [Термін придатності]
	Вивести усі заготовки зроблені до N року (N ввести з клавіатури)
10	Довідник - « Художня галерея » Поля - [Назва полотна] [Автор] [Рік] [Розмір] [Ціна]
	Вивести усі полотна певного автора (автора вводити з клавіатури)

11	Довідник - « Ремон взуття » Поля - [ID взуття] [дата прийому] [Вид роботи] [Тел. власника] [Ціна]
	Вивести усі прийняті пари взуття за номером телефону власника (номер телефону вводити з клавіатури)
12	Довідник - « Розклад занять » Поля - [Група] [День тижня] [№ пари] [Аудиторія] [Предмет]
	Вивести усі пари певної групи (Групу вводити з клавіатури)
13	Довідник - « Бібліотека » Поля - [Назва] [Автор] [Видавництво] [Тираж] [Рік]
	Вивести усі книжки певного видавництва (видавництво вводити з клавіатури)
14	Довідник - « Орнітолог » Поля - [Назва виду] [Сімейство] [Кількість особин] [Регіон розповсюдження] [Середня вартість]
	Вивести усіх птахів з вартістю більше N (N ввести з клавіатури)
15	Довідник - « Вокзал » Поля - [Пункт відбуття] [Пункт Прибуття] [Маршрут] [Місце] [Вартість квитка]
	Розрахувати середню вартість квитка
16	Довідник - « Контакти » Поля - [Ім'я] [Телефон] [Вік]
	Розрахувати середній вік

Приклад виконання

Варіант – 16.

```
[35]: a="six sick hicks nick six slick bricks with picks and sticks"#вхідний рядок
      alf1="abcdefghijklmnopqrstuvwxyz"#алфавіт нижнього регістру
      alf2="ABCDEFGHIJKLMNOPQRSTUVWXYZ"#алфавіт верхнього регістру
      sp=" " #пробіл
      i=0
      a=alf2[alf1.find(a[0])] + a[1:] #зміна регістру першого символу
      while i<len(a): #доки рядок не закінчився
          if(a[i]==sp): #пошук пробілу у рядку
              a=a[:i+1]+alf2[alf1.find(a[i+1])] + a[i+2:] #зміна регістру символу після пробілу
              k=alf1.find(a[i-1])
              a=a[:i-1]+ "%d"%k + a[i:] #заміна символу до пробілу на його номер у алфавіті
              if(k>9): #перевірка чи змінився розмір рядку
                  i+=1
              i+=1
      a=a[:-2]+ "%d"%alf1.find(a[-1]) #заміна останнього символу до на його номер у алфавіті
      i=0;
      print(a)
```

Si23 Sic10 Hick18 Nic10 Si23 Slic10 Brick18 Wit7 Pick18 An3 Stic18

```
[36]: #форматування цифр
      for i in range(len(a)):
          if(alf1.find(a[i])!=-1 and alf2.find(a[i])!=-1 and sp.find(a[i])!=-1):
              print("%d" %a[i], end = '')
          else:
              print(a[i], end = '')
```

Si23 Sic10 Hick18 Nic10 Si23 Slic10 Brick18 Wit7 Pick18 An3 Stic18

```
[37]: y= [[3,4,2,5,4],\
          [4,5,8,5,7],\
          [5,4,5,6,7],\
          [3,3,5,8,8],\
          [9,4,5,6,7]] #вхідна матриця
      z=list(); #результуючий вектор
      sum=0
      for j in range(5):
          sum=0;
          for i in range(5):
              sum+=y[i][j] # підрахунок суми елементів
          z+= [sum/5]; # формування результуючого вектору
      print(z);
```

[4.8, 4.0, 5.0, 6.0, 6.6]

Рисунок 3.36 – Виконання першого завдання

```

[*]: person = {}.fromkeys(['name', 'age', 'phone'])
person_list = list()
c=-1
def age(person_list):
    i=0
    s=0
    for i in range(len(person_list)):
        s+=person_list[i]['age']
    return s/len(person_list)
def add(person_list):
    person_list+= [person.copy()]
    person_list[-1]['name']=input("Введіть ім'я: ")
    person_list[-1]['age']=int(input("Введіть вік: "))
    person_list[-1]['phone']=input("Введіть телефон: ")
def print_p(person_list):
    i=0
    for i in range(len(person_list)):
        print(person_list[i],sep="/n")
while c!=0:
    print("Меню")
    print("1. Додати запис")
    print("2. Переглянути усі записи")
    print("3. Підрахувати середній вік")
    print("0. завершити роботу")
    c=int(input());
    if c==1:
        add(person_list)
    if c==2:
        i=0
        print_p(person_list)
    if c==3:
        print("Середній вік =",age(person_list))

```

Рисунок 3.37 – Виконання другого завдання

```
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
1
Введіть ім'я: Михайло
Введіть вік: 23
Введіть телефон: 0959595955
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
1
Введіть ім'я: Петро
Введіть вік: 25
Введіть телефон: 0999995999
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
1
Введіть ім'я: Кирило
Введіть вік: 28
Введіть телефон: 0509550999
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
2
{'name': 'Михайло', 'age': 23, 'phone': '0959595955'}
{'name': 'Петро', 'age': 25, 'phone': '0999995999'}
{'name': 'Кирило', 'age': 28, 'phone': '0509550999'}
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
3
Середній вік = 25.333333333333332
..
```

Рисунок 3.38 – Результат виконання другого завдання

Контрольні запитання

1. Що таке зрізи, та як вони застосовуються у рядках та списках.
2. Які існують можливі операції з рядками.
3. Що таке конкатенація рядків, які існують способи конкатенації.
4. Які існують методи обробки рядків, наведіть приклади.
5. Форматування - як застосовується та для чого.
6. Які існують можливі операції з списками.
7. Якого типу можуть бути елементи в межах одного списку.
8. Наведіть приклади вбудованих функцій для роботи зі списками.
9. Опишіть механізми звернення, заміни та видалення елементу рядка.
10. Множення списків, як виконується та для чого.
11. Які існують можливі операції з кортежами.
12. Якого типу можуть бути ключі в межах одного словника.
13. Якого типу можуть бути значення в межах одного словника.
14. Які існують можливі операції з словниками.
15. Наведіть приклади вбудованих функцій для роботи зі словниками.
16. Опишіть механізми звернення, заміни та видалення елементу словника.
17. Вкажіть способи звернення до елементів вкладеного словника.