

# Практична робота «Реалізація алгоритмів повторення в середовищі Thonny»

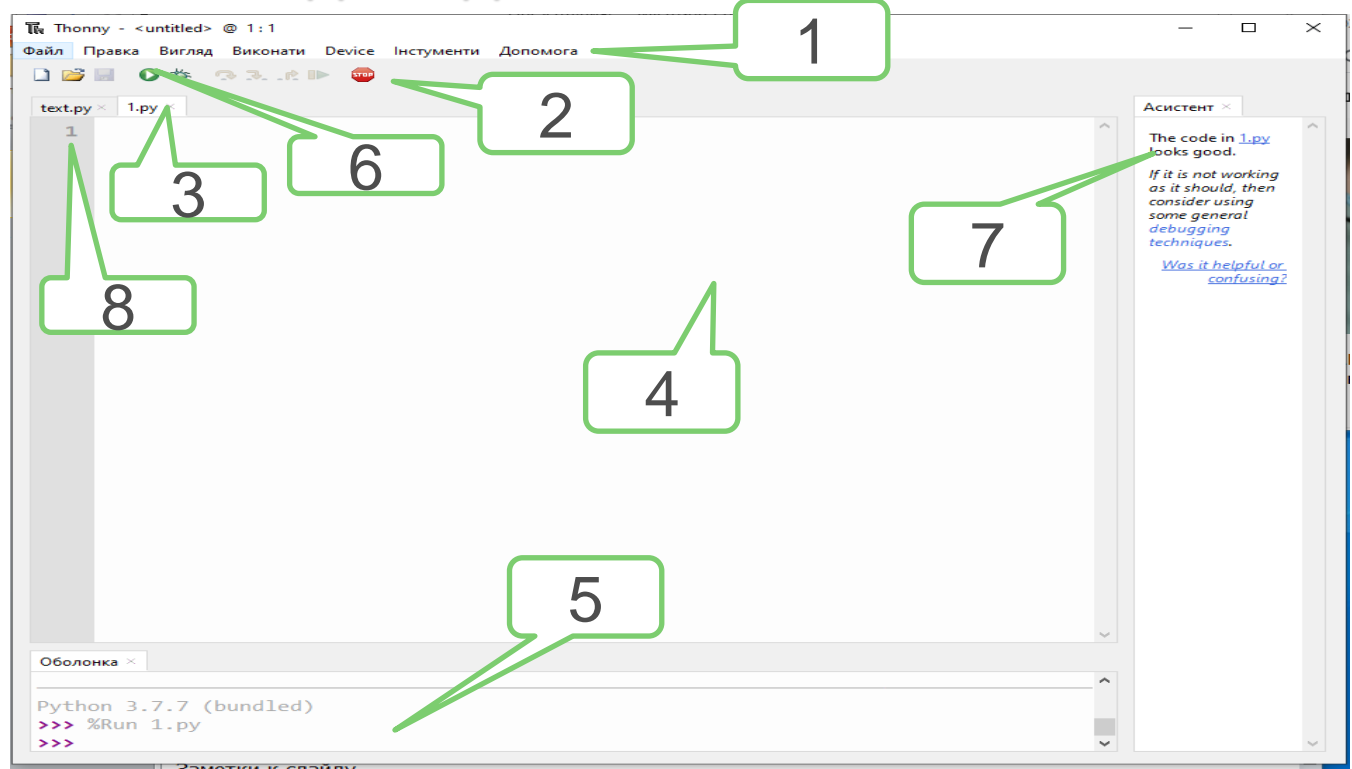
Тн



**Невдача —  
це також результат.  
Якщо ти не помиляєшся,  
ти недостатньо інноваційний**



## 1. Встановіть відповідність

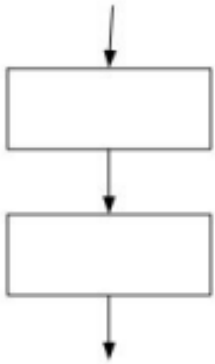


- |   |  |
|---|--|
| 1 | 1 Поле виконання програми                  |
| 2 | 2 Редактор програмного коду                |
| 3 | 3 Назва файлу                              |
| 4 | 4 Меню середовища                          |
| 5 | 5 Номери рядків                            |
| 6 | 6 Асистент (вікно повідомлень про помилки) |
| 7 | 7 Панель інструментів                      |
| 8 | 8 Запуск програми на виконання             |

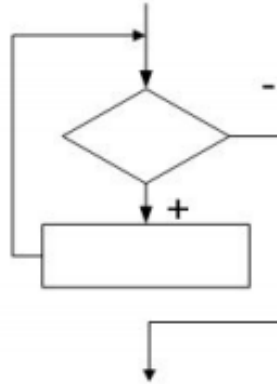




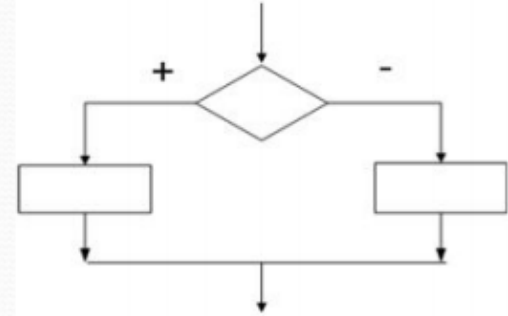
## 2. Наведіть приклади алгоритмів з власного досвіду



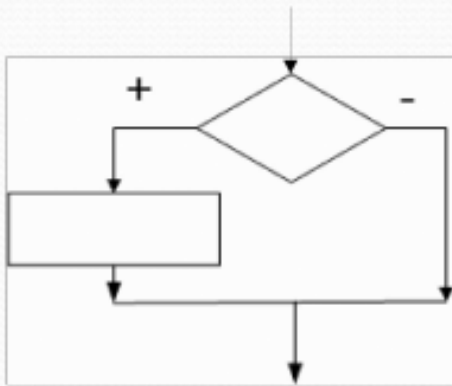
лінійний



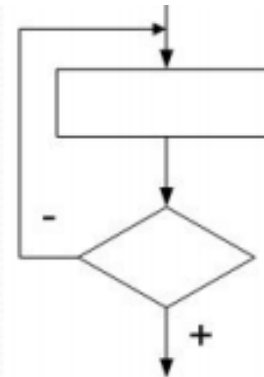
цикл  
з передумовою



розгалуження  
(повна форма)



розгалуження  
(неповна форма)



цикл  
з лічильником



### 3. Встановіть відповідність

1	частка від ділення (цілочислене ділення)	1	==
2	дорівнює	2	/
3	присвоїти	3	//
4	не дорівнює	4	%
5	остача від ділення	5	*
6	піднесення до степеня	6	!=
		7	=
		8	**

4. Оберіть базову алгоритмічну структуру для розв'язування наступної задачі:

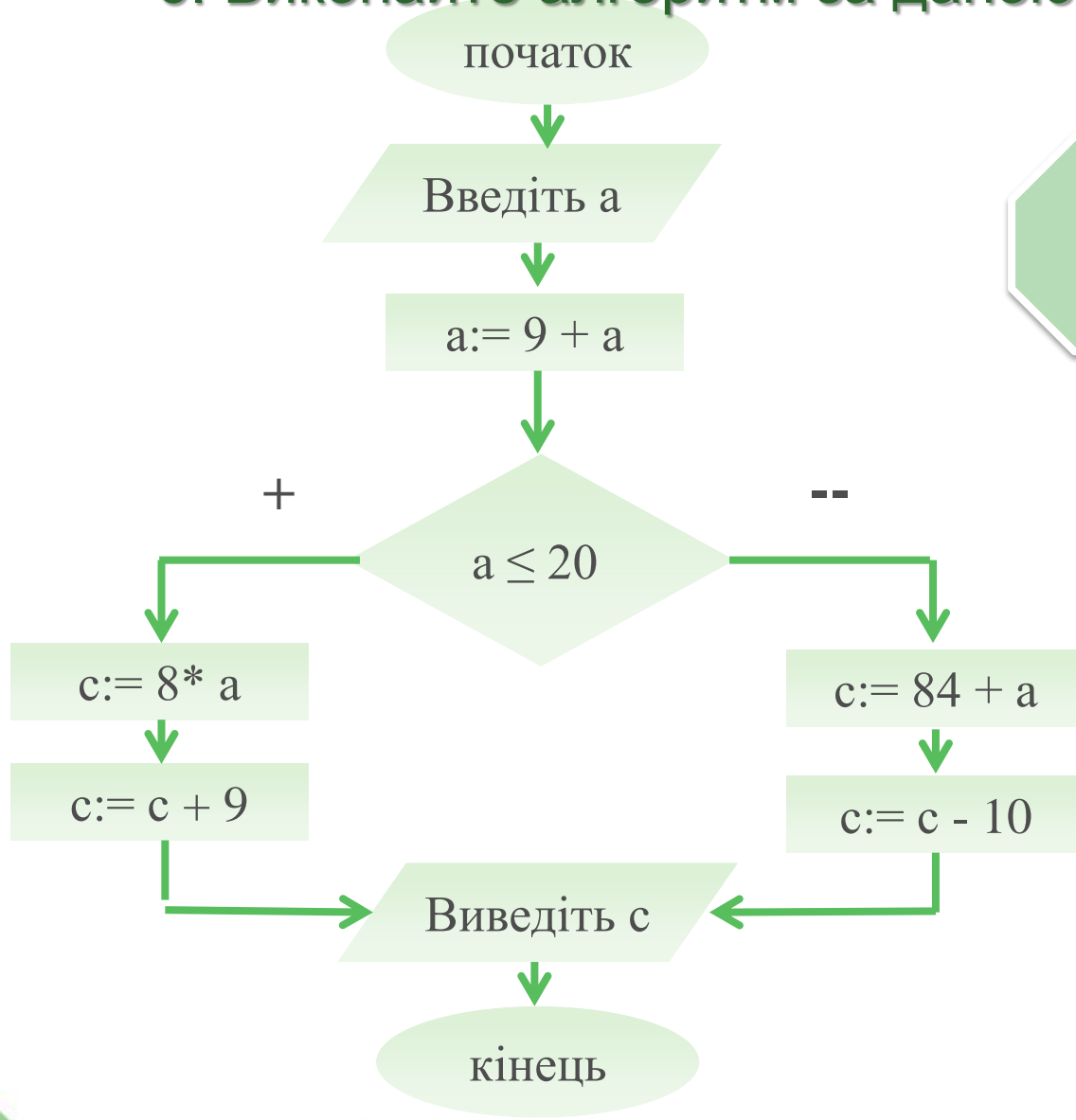
Визначити чи ділиться задане число на 13.

5. Скільки ітерацій виконається при використанні вказаного циклу:

```
for i in range(1, 26):  
    print(i**2)
```



## 6. Виконайте алгоритм за даною блок-схемою

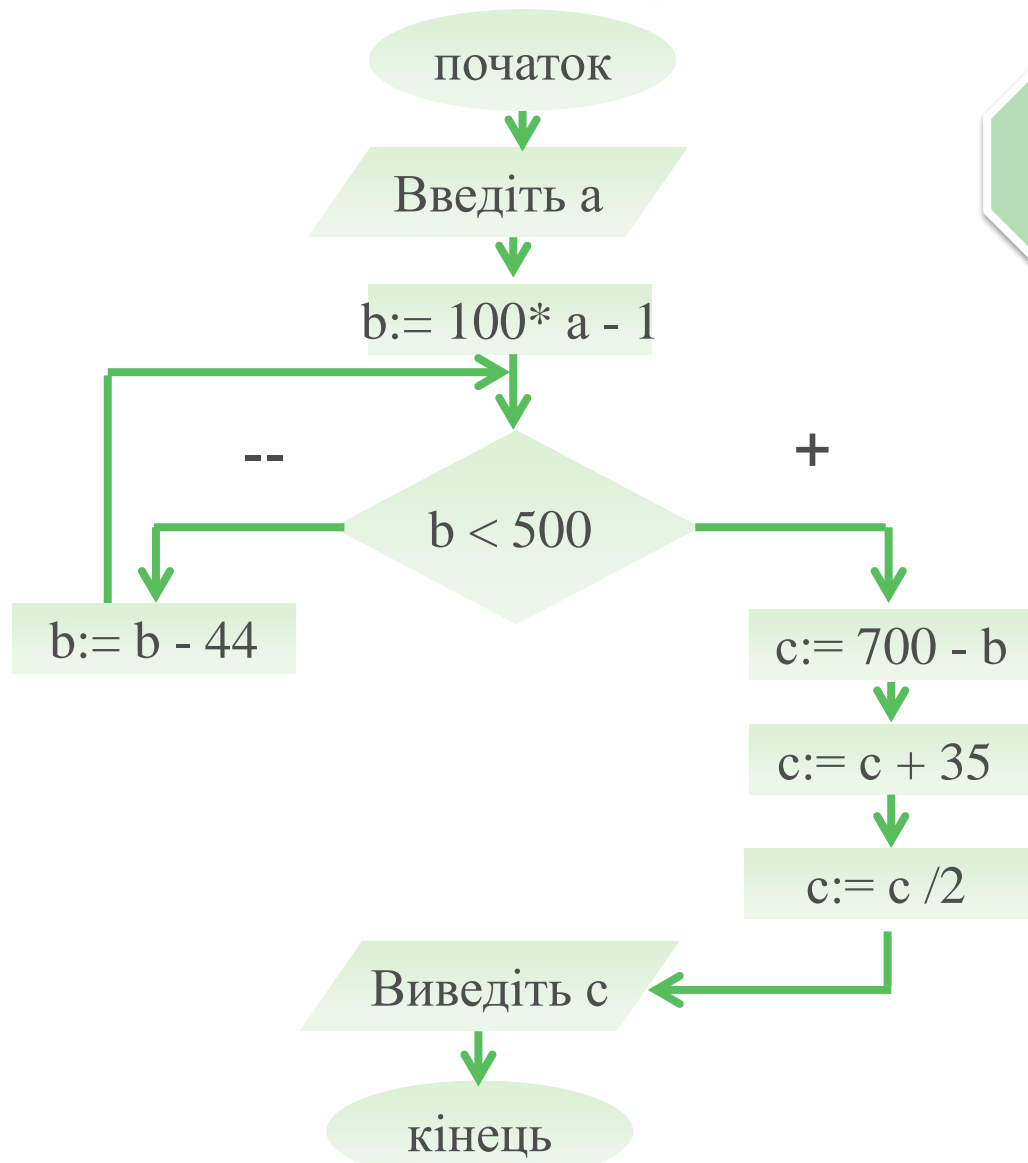


В-1  
 $a=5$

В-2  
 $a=31$



## 7. Виконайте алгоритм за даною блок-схемою



B-1  
 $a=3$

B-2  
 $a=7$





# Код не працює :-)

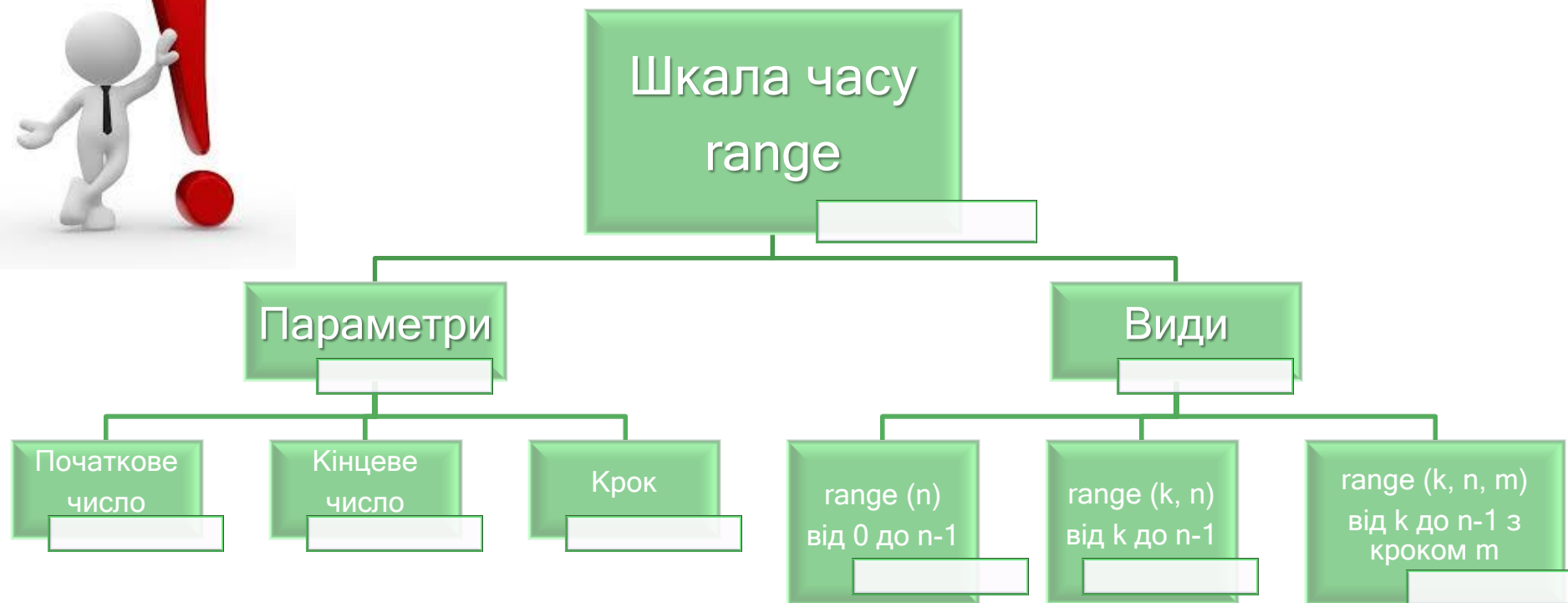


<http://pythonforbiologists.com>

Переклад Q-bart







range (100) – числа від 0...99

range (7, 15) – числа 7...14

range(1, 10, 3) – 1, 4, 7

range(3,-3,-2) – 3, 1, -1



# Під час роботи за комп'ютером дотримуйся таких правил:



**15 хв.**

Час безперервної роботи за комп'ютером має не перевищувати 15 хв



Руки мають бути чистими та сухими.



Тримай безпечну відстань очей до екрана монітора – від 50 см.



Не використовуй сторонні предмети.



Не принось на робоче місце їжу та напої.



Розміщуй ПК на рівній поверхні.



При виникненні незвичайної ситуації з комп'ютером: дивні сигнали, блимання, запах та інші неполадки в роботі – негайно повідом учителя або лаборанта у школі, й вдома – дорослих.



## Задача 1.

Дано натуральне число  $n$ . Знайти  $n!$  (добуток всіх натуральних чисел, що не перевищує  $n$ , тобто  $1*2*3*...*n$ ).

Введення числа  
з клавіатури

Надання величині  
початкового  
значення

Запуск циклу  
множення  
послідовності чисел

Виведення числа  
на екран

При множенні  
початкове значення - 1

```
1 n=int(input('n='))
2 p=1
3 for i in range(1, n+1):
4     p*=i
5 print(p)
6
```

Наперед відомо кількість повторень –  
використовуємо цикл з лічильником

```
Python 3.7.7 (bundled)
>>> %Run 1.py

n=6
720
```



## Задача 2: Знайдіть суму цифр довільного цілого числа

Наперед невідомо кількість повторень –  
використовуємо цикл з передумовою

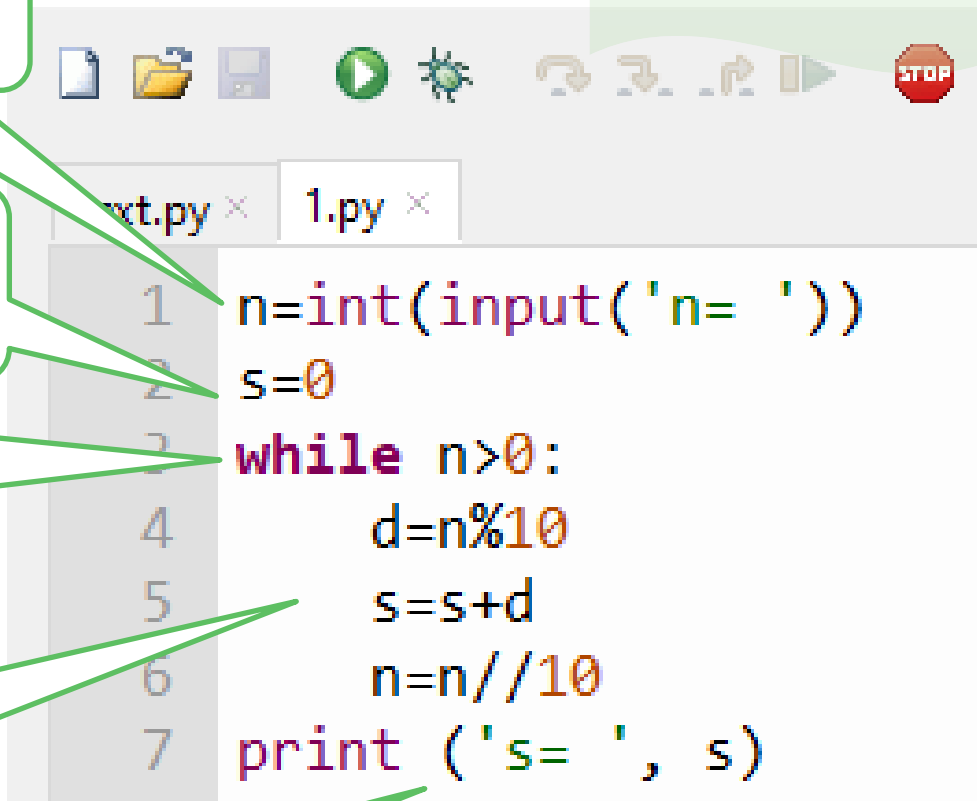
Введення числа  
з клавіатури

Надання величині  
початкового  
значення

Запуск циклу  
розбиття  
числа на цифри

Цілочисленне і  
ділення з остачею

Виведення числа  
на екран



```
1 n=int(input('n= '))
2 s=0
3 while n>0:
4     d=n%10
5     s=s+d
6     n=n//10
7 print ('s= ', s)
```

```
>>> %Run 1.py
```

```
n= 452
s= 11
```

```
>>> %Run 1.py
```

```
n=
```

Python 3.7.7 (bundled)

```
>>> %Run 1.py
```

```
n= 620
s= 8
```

```
>>> %Run 1.py
```

```
n= 6957
s= 27
```





**Задача 3. Складіть програму, яка задане ціле число перетворює на число, записане навпаки**

- 1. Чи відомо наперед кількість ітерацій?*
- 2. Як розбити число на цифри?*
- 3. Як знову зібрати цифри у зворотньому порядку?*

**Задача 4. Складіть програму, яка малює кольорову квадратну спіраль, використовуючи синій, жовтий, червоний і зелений колір**



- 1. Як викликати графічний модуль?*
- 2. Як задати кольори?*
- 3. Чи відомо наперед кількість ітерацій?*
- 4. Чому дорівнює повен оберт виконавця?*
- 5. Які команди для переміщення можна використовувати в графічному модулі?*

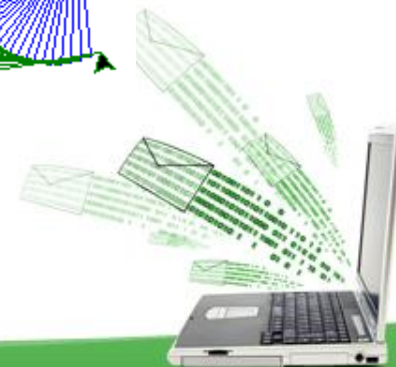
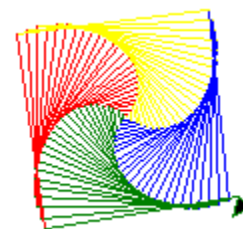


Задача 3. Складіть програму, яка задане ціле число перетворює на число, записане навпаки

```
text.py × 1.py ×  
1 n1 = int(input("Введіть ціле число: "))  
2 n2 = 0  
3 while n1 > 0:  
4     digit = n1 % 10  
5     n1 = n1 // 10  
6     n2 = n2 * 10  
7     n2 = n2 + digit  
8 print('Обернене йому число:', n2)  
9
```

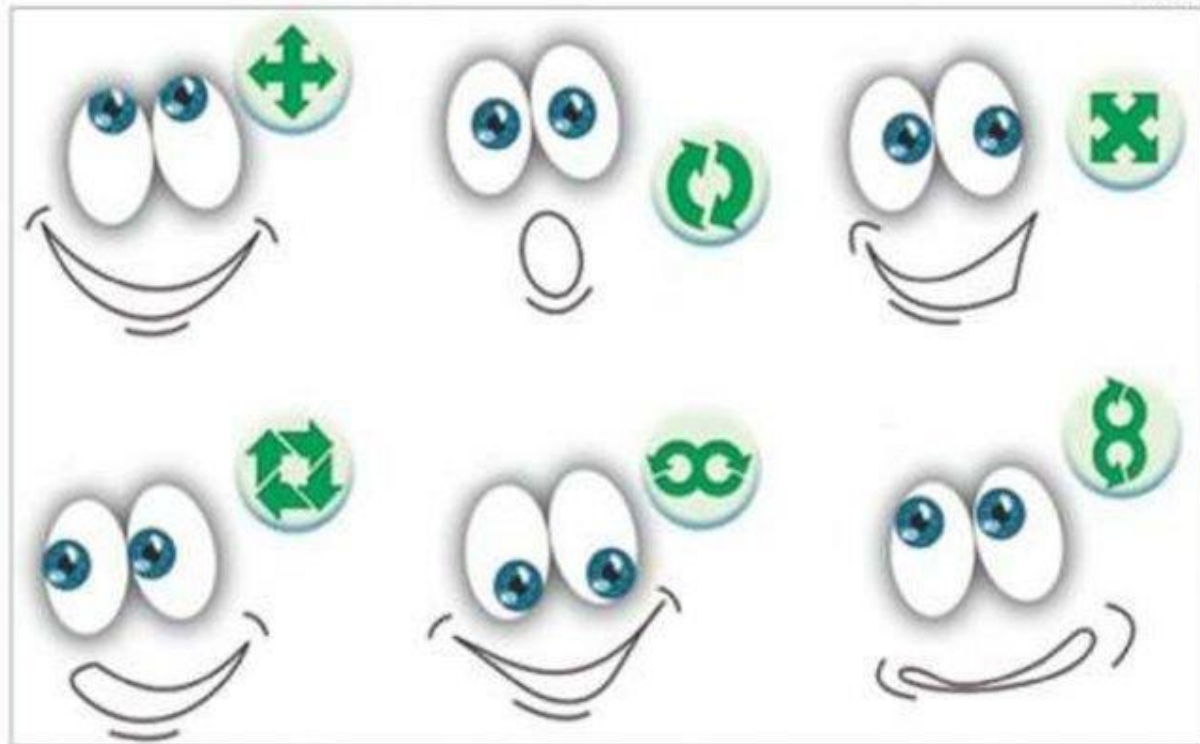
Задача 4. Складіть програму, яка малює кольорову квадратну спіраль

```
text.py × 1.py ×  
1 import turtle  
2 t = turtle.Pen()  
3 colors = ["blue", "yellow", "red", "green"]  
4 for x in range(100):  
5     t.pencolor(colors[x%4])  
6     t.forward(x)  
7     t.left(91)  
8
```



# Фізкультхвилинка для очей:

Рис.1



1. Що буде надруковано в результаті виконання фрагменту програми?

```
x = 0
while x<5:
    x = x+1
    print(x, end = ' ')
```

- А 0
- Б 4
- В 6
- Г 5

2. Дано циклічну конструкцію:

```
for i in range(1, 3):
    for j in range(1, 4):
        for k in range(1, 3):
            print (i, j, k)
```

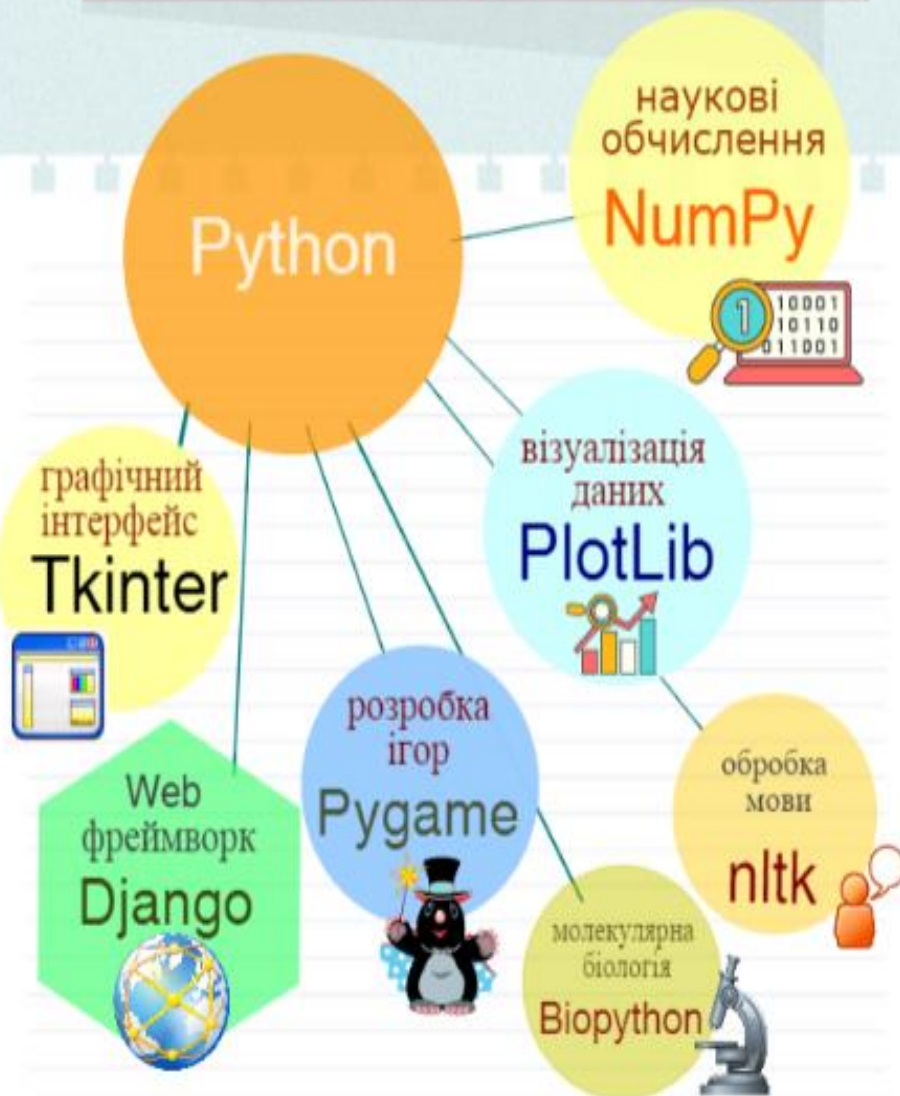
Установіть відповідність між фрагментами (1–3) і (А–В) твердженнями.

- |                          |            |
|--------------------------|------------|
| 1 Цикл for i виконається | А 6 разів  |
| 2 Цикл for j виконається | Б 2 рази   |
| 3 Цикл for k виконається | В 12 разів |





## Де використовують Python



## Домашнє завдання:

- 1). Повторити п. 12
- 2). Скласти програму визначення середнього арифметичного послідовності натуральних чисел, найбільше з яких задається з клавіатури
- 3). Скласти програму «Квадрат», записану через цикл з передумовою
- 4). Скільки чисел буде виведено у результаті виконання програми

```
for x in range (10):  
    print (x)
```

- 5) Результатом програми буде

```
k=1  
s=0  
while k<6:  
    s=s+2  
    k=k+1  
print (s)
```

- б) Як знаходити суму цифр від'ємного числа?





*Машини повинні працювати.  
Люди повинні думати  
Принцип ІВМ*