

ПРОГРАММИРОВАНИЕ НА PYTHON

ДОМАШНЕЕ ЗАДАНИЕ №2

ТЕМА: ЦИКЛЫ FOR И WHILE. ФУНКЦИЯ RANGE()

ЗАДАНИЕ

ЗАДАЧА 1

Напишите программу, которая использует ровно три цикла `for` для печати следующей последовательности символов:

KK

KK

KK

KK

KK

KK

yyyyyy

333

333

333

b

MMM

MMM

MMMM

MMM

MMM

MMM

MMM

AAAAAA

ЗАДАЧА 2

На вход программе подаётся натуральное число n , а затем n целых чисел, каждое на отдельной строке. Напишите программу, которая подсчитывает сумму введённых положительных чисел.

Пример ввода:

```
14
3
-5
1
10
-1
8
17
-98
31
2
-7
21
7
5
```

Пример вывода:

```
105
```

ЗАДАЧА 3

На вход программе подаётся натуральное число n . Напишите программу, которая вычисляет сумму всех его делителей.

Пример ввода:

```
50
```

Пример вывода:

```
93
```

ЗАДАЧА 4

На вход программе подаётся два натуральных числа n и m , каждое на отдельной строке. Напишите программу, которая печатает прямоугольник из символов '*' размерами $n \times m$.

Пример ввода:

```
3
16
```

Пример вывода:

```
*****
*****
*****
```

ЗАДАЧА 5

На вход программе в цикле подаётся последовательность целых чисел делящихся на 7, каждое число на отдельной строке. При появлении любого числа не делящегося на 7 цикл прерывается. После окончания цикла программа выводит все числа в одну строку.

Пример ввода:

```
7
7
14
21
13
```

Пример вывода:

```
7 7 14 21 13
```

ЗАДАЧА 6

Напишите программу, которая считывает два целых числа m и n ($m \leq n$) и выводит в одну строку все числа от m до n включительно.

Пример ввода:

2
10

Пример вывода:

2 3 4 5 6 7 8 9 10

ЗАДАЧА 7

На вход программе подаётся натуральное число n , а затем n целых чисел, каждое на отдельной строке. Напишите программу, которая подсчитывает сумму введенных чисел.

Пример ввода:

5
3
2
1
10
-1

Пример вывода:

15



ЗАДАЧА 8

Напишите программу, которая считывает натуральное число n и выводит первые n чисел последовательности Фибоначчи.

Последовательность Фибоначчи – это последовательность натуральных чисел, которая начинается с двух единиц, а каждое последующее число является суммой двух предыдущих: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

Пример ввода 1:

1

Пример вывода 1:

1

Пример ввода 2:

17

Пример вывода 2:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597

Сохраните код к каждой задаче в отдельный файл с расширением .py

ПОДСКАЗКИ

1. Оператор `break` осуществляет принудительный выход из цикла `for` или `while`:

```
>>> for i in range(1, 10):  
...     if i % 3 == 0:  
...         break  
...  
>>> print(i)  
3
```

2. Оператор `continue` позволяет перейти к следующей итерации цикла `for` или `while`:

```
>>> for i in range(1, 10):  
...     if i**2 % 10 == 1: continue  
...     if i**2 % 10 != 6: print(i, end=' ')  
...  
2 3 5 7 8
```

3. Если мы хотим вычислить в цикле сумму некоторых значений, то необходимо заранее объявить переменную для этого:

```
>>> summ = 0
>>> for i in range(5):
...     sum += i
...
>>> print(sum)
10
```

Оператор `+=` является расширенным оператором присвоения, который перезаписывает переменную слева, прибавляя к ней значение справа. Таким образом, запись `x += 4` эквивалентна записи `x = x + 4`

4. У функции `print` есть скрытый аргумент, которые можно изменить. Он называется `end` и отвечает за символ, который будет напечатан в конце строки – по умолчанию это `'\n'`, специальный символ переноса строки:

```
>>> print("Несколько пустых строк", end='\n\n\n')
Несколько пустых строк
```

```
>>> for i in range(10):
...     print(i, end=' ')
... print()
...
0 1 2 3 4 5 6 7 8 9
>>>
```