

Задание №2. Вариант 2

Напишите программу на языке Питон с использованием циклов.

Входные данные пользователь должен вводить с клавиатуры

1. Напишите программу, которая **по отдельности** выводит один за другим следующие рисунки. Высота треугольника (n) должна отдельно считываться с клавиатуры.

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

2. Совершенные числа
 - 2.1. Напишите **функцию**, которая принимает на вход одно число n и возвращает список делителей этого числа.
 - 2.2. Целое число называется совершенным, если сумма его делителей, включая 1 (но не само число), равна этому числу. Например, 6 является совершенным числом, поскольку $6=1+2+3$. **Используя функцию, написанную в упр. 2.1**, напишите вторую функцию, которая принимает на вход одно число и возвращает истину, если оно совершенное.
 - 2.3. **Используя функции, написанные в упр. 2.2 и упр. 2.3**, напишите программу, печатающую все совершенные числа в диапазоне от 1 до 1000. Напечатайте все делители для каждого **совершенного** числа. Программу можно написать в глобальной области.
3. Вычисление факториала. Факториалом натурального числа N называется число $N! = 1*2*3*...*N$. Написать программу, вычисляющую факториал для введенного числа N.
4. Целое число называется простым, если оно делится только на 1 и на само себя. Например, 2, 3, 5 и 7 – простые числа, а 4, 6, 8 и 9 – нет. Напишите функцию, которая принимает на вход число и возвращает истину, если оно простое и ложь иначе. Напишите программу, которая, используя данную функцию, выведет все простые числа от 1 до 1000.

Подсказки:

Вывод чисел от A до B (не включая) с шагом STEP (возможно, отрицательным):

```
for num in range(A, B, STEP):
    print(num)
```

Печать значения без перевода курсора на новую строку:

```
print("Hello, world!", end = "")
```

Вывод строки из N символов "*":

```
print(n * "*")
```

Цикл while, выводящий числа от 1 до 10:

```
i = 1
while (i <= 10):
    print(i)
    i += 1
```

Подсказки о том, как работать со списками, можно найти на сайте <http://prog.tversu.ru> в разделе Ресурсы => Краткий справочник по Питону

План выполнения упр. 1 (после каждого шага – запустить на выполнение!)

Для первой фигуры

- 1) Считать с клавиатуры число N
- 2) Написать цикл, выполняющийся N раз, и на каждой итерации печатающий «одну звездочку»
- 3) Сообразить, сколько звездочек на каждой итерации должен печатать цикл и вывести это число вместо «одной звездочки»
- 4) Заменить в теле цикла вывод «числа звездочек» на цикл, который печатает соответствующее количество звездочек

Для второй фигуры

- 1) Считать с клавиатуры число N
- 2) Написать цикл, выполняющийся N раз, и на каждой итерации печатающий «одну звездочку»
- 3) Сообразить, сколько пробелов и звездочек на каждой итерации должен печатать цикл и вывести эти два числа вместо «одной звездочки»
- 4) Заменить в теле цикла вывод «числа пробелов» и «числа звездочек» на два идущих друг за другом цикла: первый печатает соответствующее количество пробелов, второй – звездочек

План выполнения упр. 2.1 (после каждого шага – запустить на выполнение!)

- 1) Написать функцию, которая принимает на вход N и возвращает пустой список
- 2) Внутри функции написать цикл, который пробегает по всем потенциальным делителям (т.е. числам, среди которых имеет смысл искать делители) числа N и добавляет каждый из них в список. Для упрощения написания упр. 2.2 и упр. 2.3 можно не добавлять в результат само число N (хотя оно тоже является делителем N)
- 3) Перед добавлением числа в результирующий список добавить проверку условия на делимость
- 4) После цикла вернуть получившийся список

План выполнения упр. 2.2 (после каждого шага – запустить на выполнение!)

- 1) Написать функцию, которая принимает на вход N и возвращает ложь
- 2) Вызовите функцию из упр. 2.1 и сохраните результат в переменной
- 3) Напишите цикл, который пробегается по полученному списку делителей и складывает их
- 4) После цикла сравните полученную сумму со значением параметра n – если они равны, то вернуть истину, иначе – ложь

План выполнения упр. 2.3 (после каждого шага – запустить на выполнение!)

- 1) Напишите цикл, перебирающий числа от 1 до 1000
- 2) Добавьте в цикл условие. Тестом условного оператора будет значение, возвращаемое вызовом функции из упр. 2.1. Если тест истинен, напечатайте соответствующее число на экран
- 3) После печати числа (внутри условия) добавьте вызов функции из упр. 2.2 и напечатайте полученный список делителей