Основы программирования (Python)

Вывод данных:

print(“Текст сообщения”) или **print**(‘Текст сообщения’)

С переменными (обратите внимание – для присваивания используется «=», предварительного объявления переменных не требуется):

```
first_name = "ada"
last_name = "lovelace"
full_name = f"{first_name} {last_name}"
print(full_name)
```

Кстати, можно присваивать значения сразу нескольким переменным:

```
x, y, z = 0, 0, 0
```

Ввод данных пользователем:

input() – в скобках пишется поясняющий текст. Результат ввода воспринимается как строка. Например:

```
text = input(‘Напишите любое предложение: ’)
```

Если нужно, чтобы пользователь ввел число, то используем функцию **int()** преобразования строки в целое число:

```
height = input("Укажите свой рост в см:  ")
height = int(height)
```

Арифметические операции: +, –, *, /
Возведение в степень: x ** y
Получение целой части от деления: x // y
Остаток от деления: x % y

Условный оператор:

```
if <условие1>:
    <действие 1>
elif <условие2>:
    <действие 2>
else:
    <действие 3>
```

Роль операторных скобок выполняют отступы.
При проверке нескольких условий можно использовать **and** и **or**.

Цикл:

```
for value in range(start,finish):
    <действия>
```

Цикл с условием:

```
while <условие>:
    <действия>
```

Если используем логическую переменную, то ее возможные значения: **True** и **False**.

Счетчик

```
m = 0                        //инициализация счетчика
for k in range(1,N):         //в цикле проверяем некоторое условие
    if <условие>:            //если условие выполняется
       m = m + 1                  //увеличиваем счетчик на единицу
//в результате мы посчитали, сколько раз выполнялось <условие>
```

Сумма

I вариант – суммируем все элементы:

```
m = 0                        //здесь будет храниться сумма
for k in range(0,N):
    m = m + Data[k]          //прибавляем к уже имеющейся сумме очередное
                             //слагаемое
```

II вариант – суммируем только те элементы, которые удовлетворяют <условию>:

```
m = 0                  // здесь будет храниться сумма
for k in range(0,N):
    if <условие>:                //если условие выполняется,
          m = m + Data[k]              //прибавляем очередное слагаемое
```

Поиск минимального элемента

```
a_min = Data[0]       //в качестве «стартового» минимального
                      //элемента берется первый элемент
                      //или число, заведомо большее всех
                      //остальных элементов массива
for k in range(1,N):
    if Data[k] < a_min:   //если рассматриваемый элемент массива
                          //меньше текущего минимума,
                          //то он занимает его место
                          //и теперь считается минимальным
        a_min = Data[k]
```

Поиск максимального элемента

```
a_max = Data[0]           //в качестве «стартового» максимального
                          //элемента берется первый элемент массива
                          //или число, заведомо меньшее всех
                          //остальных элементов массива
for k in range(1,N):
    if Data[k] > a_max:   //если рассматриваемый элемент массива
                          //больше текущего максимума,
                          //то он занимает его место
                          //и теперь считается максимальным
        a_max = Data[k]
```

Проверка условий

Число a кратно m –

```
if a % m == 0:
```

Число a оканчивается на m –

```
if a % 10 == m:
```

Аналогично, если a оканчивается на bc - `if a % 100 == bc:`

Таким образом, чтобы «вытащить» из числа последние n цифр, нужно взять остаток от деления этого числа на 10^n.