

While loop

# Learning objectives

- 11.1.2.1 write code using a While loop
- 11.1.2.2 implement a loop algorithm according to a flowchart
- 11.1.2.3 trace program code
- 11.4.3.2 solve applied problems from various subject areas

## **Assessment criteria:**

- correctly uses while loop
- correctly uses the loop conditions



# The while Loop

With the `while` loop we can execute a set of statements as long as a condition is true.

## Example

Print `i` as long as `i` is less than 6:

```
i = 1
while i < 6:
    print(i)
    i += 1
```

**Note:** *remember to increment `i`, or else the loop will continue forever.*



# Example

For a given integer  $N$ , print out all the squares of natural numbers that do not exceed  $N$ , in ascending order.

```
1)  $n = \text{int}(\text{input}())$      $\#n=20$   
2)  $i=1$   
3)  $\text{while } i*i < n:$   
4)            $\text{print}(i*i)$   
5)            $i+=1$ 
```

<i>line</i>	<i>n</i>	<i>i</i>	<i>condition</i>	<i>output</i>
1	20	-	-	-



# The break Statement

With the `break` statement we can stop the loop even if the while condition is true:

## Example

Exit the loop when i is 3:

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```



# The continue Statement

With the `continue` statement we can stop the current iteration, and continue with the next:

## Example

Continue to the next iteration if i is 3:

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```



# The else Statement

With the `else` statement we can run a block of code once when the condition no longer is true:

## Example

Print a message once the condition is false:

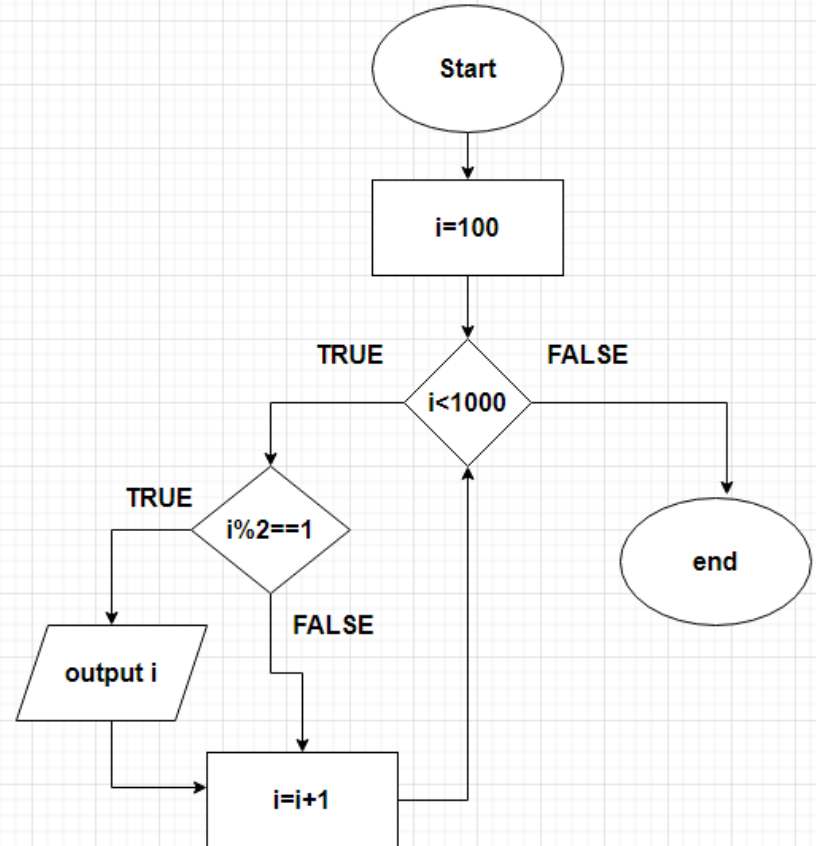
```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

# Activity

Write code for flowchart shown in picture

Descriptors:

- uses *while loop*
- uses *conditional statement*
- uses *print operator*
- writes code without error





# Task 1. Morning jog

As a future athlete you just started your practice for an upcoming event. Given that on the first day you run  $x$  miles, and by the event you must be able to run  $y$  miles. Calculate the number of days required for you to finally reach the required distance for the event, if you increase your distance each day by 10% from the previous day.

Print one integer representing the number of days to reach the required distance.

## **Descriptors:**

- solve task by using while loop
- cover given tests

# Task 1. Tests

	Input	Correct answer
Run test #1	10 20	9
Run test #2	10 30	13
Run test #3	10 100	26
Run test #4	10 10	1
Run test #5	100 101	2

## Task 2. The number of elements that are greater than the previous one

A sequence consists of integer numbers and ends with the number 0. Determine how many elements of this sequence are greater than their neighbours above.

### **Descriptors:**

- solve task by using while loop
- cover given tests

## Task 2. Tests

	Input	Correct answer
Run test #1	1 7 9 0	2
Run test #2	1 5 2 4 3 0	2
Run test #3	1 2 3 4 5 0	4
Run test #4	5 4 3 2 1 0	0

## Task 3. The second maximum

The sequence consists of distinct positive integer numbers and ends with the number 0. Determine the value of the second largest element in this sequence. It is guaranteed that the sequence has at least two elements.

### **Descriptors:**

- solve task by using while loop
- cover given tests

# Task 3. Tests

	Input	Correct answer
Run test #1	1 7 9 0	7
Run test #2	2 1 0	1
Run test #3	1 2 0	1
Run test #4	1 2 3 0	2
Run test #5	1 3 2 0	2

## Task 4. The index of a Fibonacci number

The Fibonacci sequence is defined as follows:

$$\phi_0 = 0, \phi_1 = 1, \phi_n = \phi_{n-1} + \phi_{n-2}.$$

Given an integer  $a$ , determine its index among the Fibonacci numbers, that is, print the number  $n$  such that  $\phi_n = a$ . If  $a$  is not a Fibonacci number, print -1.

### Descriptors:

- solve task by using while loop
- cover given tests

## Task 4. Test's

Input

Correct answer

Run test #1

8

6

Run test #2

10

-1

Run test #3

13

7

Run test #4

55

10

Run test #5

56

-1

Run test #6

57

-1



# Advanced level task

## В. Пицца

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Выход	стандартный вывод или output.txt

Пицца – любимое лакомство Васи, он постоянно покупает и с удовольствием употребляет различные сорта этого великолепного блюда. Однажды, в очередной раз, разрезая круглую пиццу на несколько частей, Вася задумался: на какое максимальное количество частей можно разрезать пиццу за  $N$  прямых разрезов?

Помогите Васе решить эту задачу, определив максимальное число не обязательно равных кусков, которые может получить Вася, разрезая пиццу таким образом.

**Входные данные:** содержит натуральное число  $N$  – число прямых разрезов пиццы ( $N \leq 1000$ ).

**Выходные данные:** выведите ответ на задачу.

**Примеры:**

№	Входные данные	Выходные данные
1	2	4
2	3	7

# Advanced level task

## С. Вклад

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Дано целое число  $N$ . Используя операцию деление нацело и взятие остатка от деления, найти число, полученное при прочтении числа  $N$  справа налево.

**Входные данные:** Целое число  $N$  ( $1 \leq N \leq 100000$ )

**Выходные данные:** Выведите ответ к задаче.

**Примеры:**

№	Входные данные	Выходные данные
1	125	521
2	32	23
3	156975	579651

# Advanced level task

## D. НОД

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Даны целые положительные числа  $A$  и  $B$ . Найти их наибольший общий делитель(НОД), используя алгоритм Евклида.  $\text{НОД}(A,B)=\text{НОД}(A,A \bmod B)$ , если  $B$  не равно 0;  $\text{НОД}(A,0)=A$ , где "mod" обозначает операцию взятие остатка от деления.

**Входные данные:** Целые положительные числа  $A$  и  $B$ .

**Выходные данные:** Выведите НОД  $A$  и  $B$ .

**Примеры:**

№	Входные данные	Выходные данные
1	1 2	1
2	3 2	1
3	15 75	15

# Advanced level task

## Ф. Коридор

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На этаже  $2 \cdot N$  ( $N < 100$ ) одинаковых комнат, расположенных вдоль коридора одна напротив другой. Если идти по коридору от входа, то с левой стороны номера комнат возрастают от 1 до  $N$ , а с правой- убывают от  $2 \cdot N$  до  $N+1$ . Определить номер комнаты, расположенной напротив комнаты с номером  $K$ .

**Технические требования:** Числа  $N$  и  $K$  задаются с клавиатуры. Результат выводится на экран.

**Пример:**  $N=6$   $K=3$  Результат: 10

# Advanced level task

## H. НОК

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Требуется написать программу, определяющую наименьшее общее кратное (НОК) чисел  $a$  и  $b$ .

Пример:

№	Ввод	Вывод
1	36 17	612
2	39 65	195

### Формат ввода

В единственной строке записаны два натуральных числа  $A$  и  $B$  через пробел, не превышающих 46340.

### Формат вывода

В единственную строку нужно вывести одно целое число — НОК чисел  $A$  и  $B$ .