SCHOOL
HIGHER
MEDIA
ART
AND DESIGN
POLYTECHNIC
FROM PORTO

ALP

EXERCISE SHEET

ACADEMIC ACTIVITY

Algorithms and data Structures

CURRICULAR UNIT

Sheet 03 – Iterative Structures

TOKEN

1. Write a program that **reads 10 numbers** and at the end indicates the **largest** and the **average**.

C:\WINDOWS\py.exe

```
Indique um número: 2
Indique um número: 4
Indique um número: 6
Indique um número: 8
Indique um número: 10
Indique um número: 1
Indique um número: 3
Indique um número: 5
Indique um número: 7
Indique um número: 9
A média é 5.5
o maior é 10
```

Tip! If you need to initialize a variable with the smallest possible number, you can use the math.inf function from the math library, as in the example below.

```
# Ler 10 números e indicar o maior e a média
import math

maior = -math.inf   # inicializar a variavel maior com o menor numero
```

2. Adapt the previous program, so that the user indicates in advance how many numbers he wants to read (instead of always 10 numbers).

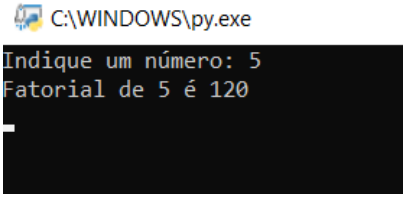
C:\WINDOWS\py.exe

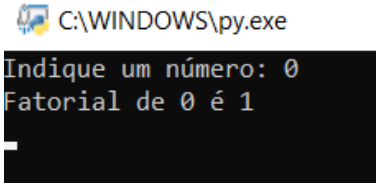
```
quando números deseja ler? 5
Indique um número: 10
Indique um número: 50
Indique um número: 40
Indique um número: 20
Indique um número: 30
A média é 30.0
o maior é 50
```

3. Create a program that simulates the **fatorial function**, that is, that determines the factorial of a given number.

   Example: Fatorial of 5 = 5 * 4 * 3 * 2 * 1 = 120
   Note that 0! = 1

   Note: do not use the math.factorial() function
   The goal is to develop our own factorial function.





4. Game **Guess the number!**
   Create a program that simulates the game of guessing a number.

   The program must start by generating a random number (between 1 and 50), allowing the player to iteratively try to guess the number.

   To generate a random number, use the random library (import random).
   In this library you will find two functions to generate random numbers:
   - random.randrange(linInf, limSup)
   - random.randint(limInf, limSup)

```
import random

num= random.randrange(0,10)  # Return random integer in range [a, b[, excludes the end points
num= random.randint(0,10)    # Return random integer in range [a, b], including both end points
```

   The player has several attempts to guess the number, and after each attempt a message like this should appear:

   - "**The number is greater**" - if the player's guess is lower than the number to be guessed
   - "**The number is smaller**" - if the player's guess is higher than the number to be guessed
   - "**Got it right**!!!" - if the player's guess matches the number to be guessed. In this case, the game ends, with the message "**Congratulations, you got it right**!!"

   Other considerations:

   • After 10 failed attempts, the game must end, indicating the player's failure., like a message like "**You have run out of 10 attempts** :(".

   • When the player guesses the number correctly, the game must indicate the **number of attempts** the player needed to get it right.

C:\WINDOWS\py.exe

```
Indique o seu palpite: 25
O número a adivinhar é MENOR

Indique o seu palpite: 12
O número a adivinhar é MENOR

Indique o seu palpite: 6
O número a adivinhar é MAIOR

Indique o seu palpite: 9
Parabéns! Acertou em 4 tentativas
```

C:\WINDOWS\py.exe

```
Indique o seu palpite: 21
O número a adivinhar é MAIOR

Indique o seu palpite: 22
O número a adivinhar é MAIOR

Indique o seu palpite: 23
O número a adivinhar é MAIOR

Indique o seu palpite: 24
Esgotou as 10 tentativas! :(
```

5. Make a 2.0 version of the previous game where:

   After completing a game, the user should be given the option to start a new game: "New game (Y/N)?".
   The program must behave according to the answer given by the user (Y or N).

6. Create a program that allows you to generate a random number between 1900 and 2020, a number that represents a year. Considering the randomly generated year, the program is intended to indicate whether the year is a leap year (*ano bissexto*) or not.

   Leap Year Definition:

   A year is a leap year if it is divisible by 4, except if, in addition to being divisible by 4, it is also divisible by 100. In this case, the year is only a leap year if it is also divisible by 400.

   In short:

   - All years that are multiples of 400 are leap years.e.g.: 1600, 2000, 2400, 2800...

   - All multiples of 4 are leap years and not multiples of 100.e.g.: 1996, 2004, 2008, 2012, 2016…

   - All other years are not leap years.

   Source:https://pt.wikipedia.org/wiki/Ano_bissexto

   After showing the result, the algorithm must ask the user if he want to generate another random number or not, acting according to the user's response.

7. Create a program that reads a number (integer and positive) and indicates whether it **is prime or not.**

   Note: A prime number is divisible only by itself and 1.

Números Primos entre 1 e 1000

Entre 1 e 1000 há 168 números primos, são eles:

2	3	5	7	11	13	17	19	23	29	31	37	41	43
47	53	59	61	67	71	73	79	83	89	97	101	103	107
109	113	127	131	137	139	149	151	157	163	167	173	179	181
191	193	197	199	211	223	227	229	233	239	241	251	257	263
269	271	277	281	283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409	419	421	431	433
439	443	449	457	461	463	467	479	487	491	499	503	509	521
523	541	547	557	563	569	571	577	587	593	599	601	607	613
617	619	631	641	643	647	653	659	661	673	677	683	691	701
709	719	727	733	739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863	877	881	883	887
907	911	919	929	937	941	947	953	967	971	977	983	991	997

```
c:\Users\mario\OneDrive\AED\4 - Exercicios\Fi
Número:23
O numero 23 é primo
Press any key to continue . . .
```

8. Create a program that illustrates the first n terms of the **Fibonacci sequence**, with the number of desired terms (n) being indicated by the user.

   Fibonacci sequence:

   In the Fibonacci sequence, each term results from the sum of the previous two.
   Source:http://pt.wikipedia.org/wiki/N%C3%BAmero_de_Fibonacci

Os **números de Fibonacci** são, portanto, os números que compõem a seguinte sequência (sequência A000045 na OEIS):

0,1, **1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...** .[nota 1][2].

   Example:

```
C:\WINDOWS\py.exe
Nº de termos a imprimir:10
Primeiro 10 termos da sequência de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34
```

9. Write a program that checks whether a given number (integer and positive) is **perfect.**

   In Mathematics, a perfect number is an integer for which the sum of all its proper positive divisors is equal to the number itself.

   For example, the number 6 is a perfect number because:

   6 is divisible by: 1, 2 and 3 1+2+3 = 6, so it is a perfect number

Os quatro primeiros números perfeitos são:

✓ $6 = 1 + 2 + 3$
✓ $28 = 1 + 2 + 4 + 7 + 14$
✓ $496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$
✓ $8128 = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 127 + 254 + 508 + 1016 + 2032 + 4064$

C:\WINDOWS\py.exe

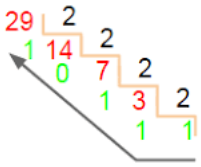
```
Indique um número:496
O número 496 é um número perfeito
```

10. Implement a program that reads a number (between 1 and 99) and determines its representation in binary language.

Example:

Number: 12 Result: 1 1 0 0

Number: 29 Result: 1 1 1 0 1



29 Decimal = 11101 Binário

11. Given a set of n numbers (n indicated by the user) integers and positives, determine the second largest value among the set of numbers read.