

# PE1: PE of 09-11-2018

---

## Master in Informatics and Computing Engineering Programming Fundamentals Instance: 2018/2019

*Here you have a possible solution for each question of the Practical on computer Evaluation.*

### 1. Integers

Write a Python program that takes the list of integers `ints = [1, 2, 2, 3, 5, 9, 13, 21, 34]` and a number `num` given by user input, and separates the list in two strings: one with the numbers less than `num` and the other with the numbers greater than `num`.

For example:

- for `num=3` the first line of the output is `"Less: 1 2 2"` and a second line with `"Greater: 5 9 13 21 34"` (without quotes)
- for `num=9` the output is `"Less: 1 2 2 3 5"` and a second line with `"Greater: 13 21 34"` (without quotes)

Save your program in the file `question1.py` inside the folder `PE1`.

Solution:

```
ints = [1, 2, 2, 3, 5, 9, 13, 21, 34]

num = int(input("Input number num: "))

less = ""
greater = ""

for n in ints:
    if n < num:
        less += str(n) + " "
    elif n > num:
        greater += str(n) + " "
```

## 2. Pairs

Write a Python program that, given a non negative integer **number** by user input, computes the number of two consecutive digits (pairs) that are equal.

For example:

- for number=1122234997550 the output is 5 (the pairs are 11, 22, 22, 99 and 55)
- for number = 1988887 the output is 3 (the pairs are 88, 88 and 88)

Save your program in the file **question2.py** inside the folder **PE1**.

Solution:

```
number = int(input("Input a non negative integer: "))

temp = number
consecutive_pairs = 0
while temp > 10:
    if (temp % 100) // 10 == temp % 10:
        consecutive_pairs += 1
    temp //= 10

#print("Number = ", number)
#print("Consecutive pairs = ", consecutive_pairs)
print(consecutive_pairs)
```

### 3. Sequences

Write a Python program that has two lists of equal size referenced by variables `integers` and `reals`, with values of your choice. The program prints the sequence with the larger of the two numbers in the corresponding positions; when the numbers are equal, the result is the integer.

For example:

- for `integers = [0, 2, 9, 15, 64]` and `reals = [0.0, 3.2, 8.4, 15.5, 128.0]`, the output is the string `"0 3.2 9 15.5 128.0"` (without quotes)
- for `integers = []` and `reals = []`, the output is the empty string

Save your program in the file `question3.py` inside the folder `PE1`.

Solution:

```
integers = [0, 2, 9, 15, 64]
reals     = [0.0, 3.2, 8.4, 15.5, 128.0]

result = ""
i = 0
for n in integers:
    j = 0
    for r in reals:
        if i == j:
            if n >= r:
                result += str(n) + " "
            else:
                result += str(r) + " "
        j += 1
    i += 1

print(result)
```

## 4. Rock-Paper-Scissors

Write a Python program to implement the two-player Rock-Paper-Scissors game. Each player chooses **rock**, **scissors**, or **paper**. The program ask the input of player A, the input of player B, compare them, print out a message of congratulations to the winner.

Consider the rules: **rock** beats **scissors**, **scissors** beats **paper**, **paper** beats **rock**.

For example:

- for A=**rock** and B=**paper**, the output is **The winner is B**
- for A=**scissors** and B=**paper**, the output is **The winner is A**
- for A=**paper** and B=**paper**, the output is **That's a draw**

Save your program in the file **question4.py** inside the folder **PE1**.

Solution:

```
a = input("Choose 'rock' or 'paper' or 'scissors': ")
b = input("Choose 'rock' or 'paper' or 'scissors': ")

if a == "rock":
    if b == "scissors":
        print("The winner is A")
    elif b == "paper":
        print("The winner is B")
    else:
        print("That's a draw")
elif a == "scissors":
    if b == "paper":
        print("The winner is A")
    elif b == "rock":
        print("The winner is B")
    else:
        print("That's a draw")
else: # a == "paper"
    if b == "rock":
        print("The winner is A")
    elif b == "scissors":
        print("The winner is B")
    else:
        print("That's a draw")
```

## 5. Binary converter

Write a Python program that converts a decimal number (base 10) **dec**, given by user input, into a binary number (base 2). Decimal numbers of base 10 numbers can use only digits from 0 to 9 inclusive. Binary numbers can use digits from 0 to 1 inclusive.

For example:

- for **dec=9** the output is the binary number **1001**
- for **dec=64** the output is the binary number **1000000**
- for **dec=234** the output is the binary number **11101010**

Save your program in the file **question5.py** inside the folder **PE1**.

Solution:

```
dec = int(input("Decimal number = "))

binary = 0
aux = dec
place = 0
while aux:
    binary = aux % 2 * 10 ** place + binary
    aux //= 2
    place += 1

print(str(binary))
```

**The end.**

*FPRO, 2018/19*