WEDNESDAY

EXERCICE 1

WHAT YOUR PROGRAM SHALL DO

1. The program asks user to enter a word containing many minus (‘-‘) :

    Enter a word : r-o-na-n

2. The program will print the same word without all minus (‘-‘) :

    Word without minus : ronan

3. The program will ask user whether to continue or not: if yes, program restarts from step 1

    Do you want to continue (Y/N)? : Y

Do perform this exercise you need to use the function already defined for you (see code below)

Function name	removeMinuses
Parameters	word (a **string**)
Return value	word (a **string**)
Examples	removeMinuses (“to-to”) → toto

You need to use this function in your program:

```
def removeMinuses(word):
    newWord = ""
    for i in range(len(word)):
        char = word[i]
        if char != "-":
            newWord += char
    return newWord
```

Correction:

```
toContinue = True
while toContinue :
  word = str(input("Enter a word= "))
  print(removeMinuses(word))
  yesNo = str(input("Do you want to continue (Y/N)?= "))
  toContinue = (yesNo == "Y")
```

EXERCICE 2

WHAT YOUR PROGRAM SHALL DO

First you will to implement the following function :

Function name	sum
Parameters	number1 (an **integer**) , number2 (an **integer**)
Return value	The sum of number1 and number2 (an **integer**)
Examples	sum (2, 3) → 5

Then code the main program:

4. The program asks user to enter 2 numbers :

```
Number 1  : 12
Number 2  : 13
```

5. The program will print the sum of the 2 numbers

```
The sum is  : 25
```

Warning : you need to call the function you have defined previously

Correction:

```
def sum(x, y):
  return x + y

nbr1 = int(input("Number 1: "))
nbr2 = int(input("Number 2: "))
print("The sum is :", str(sum(nbr1,nbr2)))
```

EXERCICE 3

WHAT YOUR PROGRAM SHALL DO

You shall use the same function **sum()** previously created, with this program:

6. The program asks user to enter the number of values:

```
Number of values : 3
```

7. Then the program asks user to enter each values one by one:

```
Value 1 : 10
Value 2 : 5
Value 3 : 2
```

8. The program will print the sum of all numbers

```
The sum is  : 17
```

Warning : You cannot make a sum directly, you need to call the function you have defined previously

Correction:

```
def sum(x, y):
  return x + y

nbrValues = int(input("Number of values: "))
result = 0
for i in range (nbrValues):
  value = int(input("Value "+str(i+1)+" : "))
  result = sum(result, value)

print("The sum is : ", result)
```

THURSDAY

EXERCICE 1

WHAT YOUR PROGRAM SHALL DO

First you need to implement the following function :

Function name	sumFromTo
Parameters	start (an **integer**) , end (an **integer**)
Return value	The sum of numbers from start to end values
Examples	sumFromTo (2, 5) → 14 Explanation : we start from 2 and we ends at 5 : 2 + 3 + 4 + 5 = 14
Warning	If start value is lower than end value, **you need to return 0**

Then code the main program:

9. The program asks user to enter the start value and the end value :

```
Start value  : 2
End value  : 5
```

10. The program will print the sum of numbers between start and end values

```
The sum of numbers between 2 and 5 is  : 14
```

Warning : you need to call the function you have defined previously

Correction:

```
def sumFromTo(x, y):
  result = 0
  for i in range (x, y+1):
    result = result + i
  return result

start = int(input("Start value : "))
end = int(input("end value : "))
print(sumFromTo(start,end))
```

EXERCICE 2

WHAT YOUR PROGRAM SHALL DO

First you need to implement the following function :

Function name	numberOfUpperCases
Parameters	word (an **string**)

Return value	The number of uppercase characters in the word (**an integer**)
Examples	numberOfUpperCases (“Phnom Pen”) → 2 *Phnom Pen has 2 uppercase characters Phnom Pen*

Then code the main program:

11. The program asks user to enter a word :
    Word : RonaN
12. The program number of uppercase characters in the word:
    Number of uppercase letters : 2

Warning : you need to call the function you have defined previously

Correction:

```
def numberOfUpperCases(text):
  total = 0
  for i in range (len(text)):
    if text[i].upper() == text[i]:
      total = total + 1
  return total

word = str(input("Word : "))
print("Number of uppercases letters : ",numberOfUpperCases(word))
```

FRIDAY

EXERCICE 1

This code does not work when running program

- Find the reason and fix it

```
# Return ‘Good’ is the grade is greater than 10
# Return ‘Bad’ is the grade is less or equal than 10
def getComment(grade):
    if grade > 10:
        return “Good”

print(getComment(12) + getComment (8))
```

Correction:

```
def getComment(grade):
  if grade > 10:
    return "Good"
  else:
    return "Bad"

print(getComment(12) + getComment (8))
```

EXERCICE 2

We have a program to display the price of banana, apple and orange. This code does not work when running program

- Find the reason and fix it

```
# banana -> 2 $
# apple  -> 5 $
# orange -> 1 $
def getPrice(fruitName):
    if fruitName == "banana":
        return 2
    if fruitName == "apple":
        return 5

print("banana price is: " + str(getPrice("banana")) + " dollars")
print("orange price is: " + str(getPrice("orange")) + " dollars")
```

Correction:

```
def getPrice(fruitName):
  if fruitName == "banana":
    return 2
  elif fruitName == "apple":
    return 5
  elif fruitName == "orange":
    return 1

print("banana price is: " + str(getPrice("banana")) + " dollars")
print("orange price is: " + str(getPrice("orange")) + " dollars")
```

EXERCICE 3

We have a program to display the absolute value of a number (ex: getAbsolute(-5) = 5)

This code does not work when running program

- Find the reason and fix it

```
def getAbsolute(number):
    if number < 0:
        return -1 * number
    else:
        return str(number)

print(getAbsolute(5) + 10)
```

Correction:

```
def getAbsolute(number):
  if number < 0:
    return -1 * number
  else:
    return number

print(getAbsolute(5) + 10)
```

SATURDAY

EXERCICE 1

We want to improve the following program to avoid duplication of code (in red):

```
# Test 1
number1 = 20
number2 = 100
result = 0
if number1 > number2:
    result = number1
else:
    result = number2
print("Maximum is " + str(result))

# Test 2
num1 = 200
num2 = 300
result = 0
if num1 > num2:
    result = num1
else:
    result = num2

print("Maximum is " + str(result))
```

1 – Implement the following function:

Function name	max
Parameters	number1 (an **integer**) , number2 (an **integer**)
Return value	The max of number1 and number2 (a **integer**)
Examples	max (2, 5) → 5

2 – Change the above code to use your function and void duplication of code (in red)

Correction:

```
def max(x, y):
  if x > y:
    return str(x)
  return str(y)

number1 = 20
number2 = 100
result = 0

print("Maximum is " + max(number1, number2))

num1 = 200
num2 = 300
result = 0

print("Maximum is " + max(num1, num2))
```

EXERCICE 2

We want to improve the following program to avoid duplication of code (in red):

```
# Test 1
text1 = "Hello PNC"
result = ""
lastIndex = len(text1) - 1
for i in range(len(text1)):
    result += text1[lastIndex - i]
print(result)

# Test 2
text2 = "Welcome 2021"
result = ""
lastIndex = len(text2) - 1
for i in range(len(text2)):
    result += text2[lastIndex - i]
print(result)
```

1 – Implement the following function:

Function name	reverseString
Parameters	word (an **string**)
Return value	The same string but characters are in the reversed order
Examples	reverseString (“ronan”) → “nanor”

2 – Change the above code to use your function and void duplication of code (in red)

Correction:

```
def reverseString(word):
  lastIndex = len(word) - 1
  result = ""
  for i in range(len(word)):
    result += word[lastIndex - i]
  return result

text1 = "Hello PNC"
print(reverseString(text1))

text2 = "Welcome 2021"
print(reverseString(text2))
```