WEEK 09

AUG 10 to AUG 14

PNC ALGORITHM CERTIFICACTION

OPTAIN YOUR CERTIFICATION ON CHAPTER 4 : *'BUILD AN ALGORITHM TO SOLVE A PROBLEM'*

MONDAY

Q1 *To help you*

WHAT YOUR PROGRAM SHALL DO

Let's compute student certification!

- Enter a student name and final grade:
  - **CONSOLE** : console shall display Student grade:
  - User enter the student name and score separated by a ";"

- Compute the **student certification result** from his/her score using following rule:

GRADE	CERTIFICATION
90-100	PLATINIUM
80-89	GOLD
70-79	SILVER
60-69	BRONZE
50-59	BASIC
<50	NO CERTIFICATION

- Display student name and certification
  - **CONSOLE** : console shall display: <name> certification is : <certification>

EXAMPLES

INPUT	EXPLANATION
>Student grade: Ronan;99 >Ronan certification is PLATINIUM	
>Student grade: Rady;50 >Rady certification is BASIC	

TIPS

Here we suppose the input is correct : a text followed by a ";" and a number

Correction

```
studentGrade = str(input("Student grade: "))
length = len(studentGrade)
findSemi = 0
result = ""

# Find semicolumn position
for i in range(length):
    if studentGrade[i] == ";":
        findSemi = i
studentName = studentGrade[0:findSemi]
studentScore = int(studentGrade[findSemi + 1: ])

# Find semicolumn position
if studentScore < 50:
    result = "NO CERTIFICATION"
elif studentScore >= 50 and studentScore <= 59:
    result = "CERTIFICATION IS BASIC"
elif studentScore >= 60 and studentScore <= 69:
    result = "CERTIFICATION IS BRONZE"
elif studentScore >= 70 and studentScore <= 79:
    result = "CERTIFICATION IS SILVER"
elif studentScore >= 80 and studentScore <= 89:
    result = "CERTIFICATION IS GOLD"
elif studentScore >= 90 and studentScore <= 100:
    result = "CERTIFICATION IS PLATINIUM"
print(studentName + " " + result)
```

Q2 (20 POINTS)

WHAT YOUR PROGRAM SHALL DO

Let's compute all student certification!
 	
- Enter the number of students
  ○ **CONSOLE** : console shall display Number of students:
 	
- For each students, enter student name and final grade:
  ○ **CONSOLE** : console shall display Student grade:
  ○ User enter the student name and score separated by a ";"
 	
- For each students, display student name and certification (same way than Q1)
  ○ **CONSOLE** : console shall display: <name> certification is : <certification>
 	
- At the end display the average final grade
  ○ **CONSOLE** : console shall display Average grade is: : <average>

EXAMPLES	
INPUT	EXPLANATION
> Number of students: 3 >Student grade: Ronan;50 >Ronan certification is BRONZE >Student grade: Rady;100 >Rady certification is PLATINIUM >Student grade: Hugo;00 >Hugo certification is NO CERTIFICATION >Average grade is: 50	The average of the3 scores is (50 +100 +0)/3 = 50

Correction:

```
numberOfStudent = int(input("Number of student: "))
allScore = 0
average = 0
for n in range(numberOfStudent):
    studentGrade = str(input("Student grade: "))
    length = len(studentGrade)
    findSemi = 0
    for i in range(length):
        if studentGrade[i] == ";":
            findSemi = i
    studentName = studentGrade[0:findSemi]
    studentScore = int(studentGrade[findSemi + 1: ])
    allScore = allScore + studentScore
    average = allScore / numberOfStudent
    result = ""
    if studentScore < 50:
        result = "NO CERTIFICATION"
    elif studentScore >= 50 and studentScore <= 59:
        result = "CERTIFICATION IS BASIC"
    elif studentScore >= 60 and studentScore <= 69:
```

```
        result = "CERTIFICATION IS BRONZE"
    elif studentScore >= 70 and studentScore <= 79:
        result = "CERTIFICATION IS SILVER"
    elif studentScore >= 80 and studentScore <= 89:
        result = "CERTIFICATION IS GOLD"
    elif studentScore >= 90 and studentScore <= 100:
        result = "CERTIFICATION IS PLATINIUM"

    print(studentName + " " + result)
print("Average grade is: ", int(average))
```

TUESDAY

Q1 *To help you*

WHAT YOUR PROGRAM SHALL DO

Revert a string!
 	
- Enter a text (string):
  ○ **CONSOLE** : console shall display Text:
- Display the reverse string (the same characters but in the reverse order)
  ○ **CONSOLE** : console shall display: Reserve text: <result>

EXAMPLES	
INPUT	EXPLANATION
>Text: ronan >Reverse text: nanor	"nanor" is "ronan" with a reversed order of letters (start from end to the beginning)
>Text: abc >Reverse text: cba	

Correction:

```
text = str(input("Text: "))
result = ""
for i in range(len(text)):
    index = len(text) - 1
    result = result + text[index - i]
print(result)
```

Q2 (20 POINTS)

WHAT YOUR PROGRAM SHALL DO
2 words have the same letters?

- Enter 2 words (string):
  - **CONSOLE** : console shall display Word 1:
  - **CONSOLE** : console shall display Word 2:

- Display "SAME-SAME BUT DIFFERENT" is the 2 words have the same letter, but not in the same order, otherwise display "DIFFERENT"

Important : for this question, all worlds shall contain **UNIQUE** characters *("abc" is good, but 'aba' is not good, since 2 letter "a")*

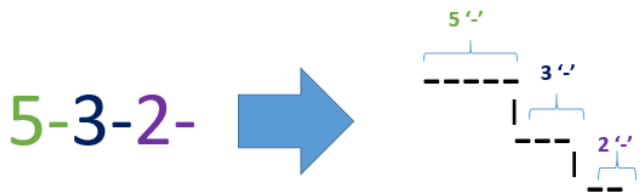
EXAMPLES	
INPUT	EXPLANATION
>Word 1: hugo >Word 2: hogu >SAME-SAME BUT DIFFERENT	All letters of word1 are in word2 All letters of word2 are in word1
>Word 1: hugo >Word 2: hogur > DIFFERENT	Word 2 contain the letter "r" and word 1 does not.

Correction:

```
wordOne = str(input("Word 1: "))
wordTwo = str(input("Word 2: "))
isTheSame = 0
for i in range (len(wordOne)):
  character = wordOne[i]
  for j in range (len(wordTwo)):
    if wordTwo[j] == character:
      isTheSame = isTheSame + 1
if isTheSame == len(wordOne) and len(wordOne) == len(wordTwo):
  print("SAME-SAME BUT DIFFERENT")
else:
  print("DIFFERENT")
```

WEDNESDAY

Q1 *20 POINTS*



WHAT YOUR PROGRAM SHALL DO
Draw a staircase! 

- Enter staircase steps width :
  - CONSOLE : console shall display : "steps:
- Each number represents a step width (number of "-")

- Print the staircase on the console
  - Each step is represented by the <width> number of "-"
  - Then add a "|" at the right position
  - Then continue with next step, at the right position

EXAMPLES	
INPUT	EXPLANATION

>Steps: 3782
--- 	
  |
  -------- 	
  |
  -------- 	
  |
  --
	The first step has 3 "-"
	The second step has 7 "-"
	Etc...

Correction:

```
steps = str(input("Steps: "))
space = ""
dash = ""
for i in range(len(steps)):
    step = int(steps[i])
    for n in range(step):
        space = space + " "
        dash = dash + "-"
    if i < len(steps) - 1:
        dash = dash +"\n" + space+ "|" + "\n" + space
    else:
        dash = dash +"\n" + space
print(dash)
```

THURSDAY

Q1 *20 POINTS*

WHAT YOUR PROGRAM SHALL DO

- Enter the number of values
  - **CONSOLE** : console shall display Number of values:

- Then enter the values
- Display those values in a more condensed way as follows:

Example of values : 2, 2, 2, 10, 6, 6, 6, 6, 2

Result :
3-2
1-10
4-6
1-2

Explanation : the values starts with 3 '2' then 1 '10' then 4 '6' and finally 1 '2'

EXAMPLES	
INPUT	EXPLANATION
>Number of values: 4 >2 >2 >2 >8 >3-2 >1-8	We have entered 3 '2' and 1 '8'

Correction:

```
numberOfvalues=int(input("Number Of values:"))
currentSize=1
currentValue=0
finishSerie = False
result=""
for index in range(numberOfvalues):
  value=int(input())
  if index==0:
    currentValue=value
  else:
    if value==currentValue:
      currentSize=currentSize+1
      finishSerie = False
    else:
      finishSerie = True
    if finishSerie == True or index == numberOfvalues-1:
      result=result+str(currentSize)+"-"+str(currentValue)+"\n"
      currentSize=1
      currentValue=value
print(result)
```

FRIDAY

Gasoline: ប្រេងសាំង

Reservoir: អាងស្តុកទឹក

Contain: ផ្ទុក, មាន

Centilitres: មិលីម៉ែត្រ

Q1 TO HELP YOU

WHAT YOUR PROGRAMM SHALL DO
We want to calculate the gasoline level of our TUKTUK Enter the TUKTUK gasoline quantity: • CONSOLE : console shall display : Gasoline quantity: Enter the list of actions : • CONSOLE : console shall display : Actions: • Enter one string (the actions): we expect it to contain only: "F" • The gasoline quantity is between 0 and 500 centiliters - if not print "INCORECT GASOLINE QUANTITY" • F mean go **Forward 50km**, for this distance we lower 120 centiliters to the reservoir • Calculate and print "The final quantity of gasoline is: <final quantity>" - If the final gasoline quantity is lower than 0 we print "IMPOSSIBLE"

EXAMPLES	
CONSOLE	EXPLANATION
> Gasoline quantity : 300 >Actions : FFF > IMPOSSIBLE	Initial quantity is 300 We entered 3 Forward 300 – 120 – 120 – 120 = **-60** Reservoir stop at **-60**, it's under than 0 so : "IMPOSSIBLE"
> Initial gasoline quantity : 600 > INCORECT GASOLINE QUANTITY	Initial quantity is 600, not correct because we expect a quantity between 0 and 500.
> Initial gasoline quantity : 450 >Actions : FF > The final quantity of gasoline is: 210	Initial quantity is 450 We entered 2 Forward 450 – 120 – 120 = **210** Reservoir stop at **210**

Q2 20 POINTS

WHAT YOUR PROGRAMM SHALL DO

(almost the same than Q1)
We want to calculate the gasoline level of our TUKTUK

Enter the TUKTUK gasoline quantity:

- CONSOLE : console shall display : Gasoline quantity:

Enter the list of actions :

- CONSOLE : console shall display : Actions:
- Enter one string (the actions): we expect it to contain only: "F" or "B" or "R"

- The gasoline quantity is between 0 and 500 centiliters
  - if not print "INCORECT GASOLINE QUANTITY"

- F means go **Forward 50km**, for this distance we lower 120 centiliters to the reservoir
- B means a **Boost of 100km**, for this distance we lower 250 centiliters to the reservoir
- R means **Reload reservoir**: the reservoir quantity Is set to 500 centiliters

- Calculate and print
  - "The final quantity of gasoline is: <final quantity>"
  - "The total distance is: <total_distance>"
    - If the final gasoline quantity is lower than 0 we print "IMPOSSIBLE"

"

EXAMPLES

CONSOLE	EXPLANATION
> Initial gasoline quantity : 420 >Actions : FBRFF > The final quantity of gasoline is: 260 > The total distance is: 250	Initial quantity is 420 Action | Gasoline quantity | Total distance F | 420 – 120 = 300 | 0 + 50 = 50 B | 300 – 250 = 50 | 50 + 100 = 150 R | 500 | 150 F | 500 – 120 = 380 | 150 + 50 = 200 F | 380 – 120 = **260** | 200 + 50 = **250** Reservoir stop at **260** Total of kilometers is **250**
> Initial gasoline quantity : 510 > INCORECT GASOLINE QUANTITY	Initial quantity is 510, not correct because we expect a quantity between 0 and 500.
> Initial gasoline quantity : 200 >Actions : FFRB > IMPOSSIBLE	Initial quantity is 200 We entered 2 Forward Action | Gasoline quantity | Total distance F | 200 – 120 = 80 | 0 + 50 = 50 F | 80 – 120 = -40 IMPOSSIBLE | Here the reservoir value become negative: "IMPOSSIBLE"

Correction:

```
gas = int(input("Gasoline quantity: "))
distance = 0
stayPositive = True
if gas>=0 and gas<=500:
  action = input("Action: ")
  for n in range(len(action)):
    if action[n]=="F":
      gas = gas - 120
      distance = distance + 50
    elif action[n]=="B":
      gas = gas - 250
      distance = distance + 100
    elif action[n]=="R":
      gas = 500
    if gas<0:
      stayPositive = False
  if not stayPositive:
    print("Impossible")
  else:
    print("The final quantity of gasoline is: ",gas)
    print("The total distance is: ",distance)
else:
  print("Incorect gasoline quentity")
```