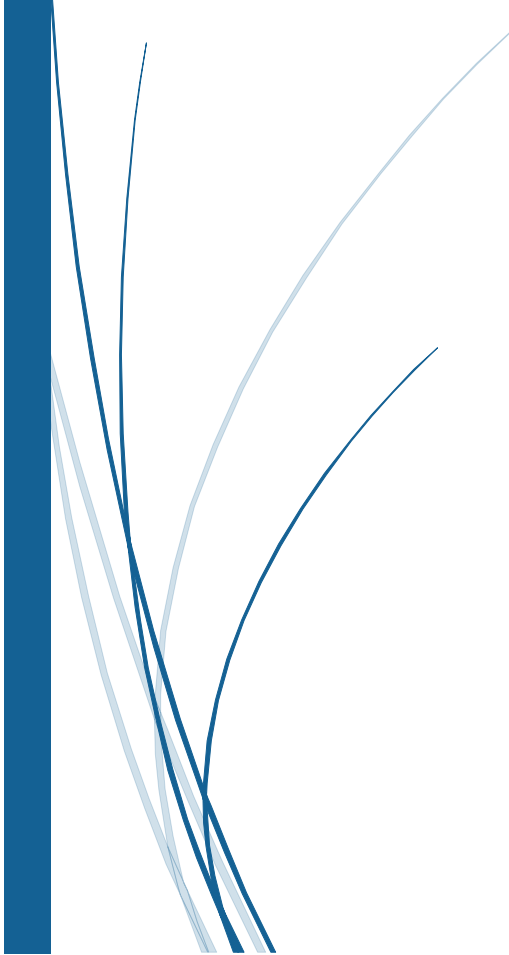




RAMONA GEORGIANA GHINEA

2009126



Software Development
Assessment 1
ACCB4017

Lecturer: Renuga Jayakumar
Date: 02 April 2021

Table of Contents

Introduction.....	2
Software Development.....	3
PART A.....	3
A1. VARIABLES	3
Definition	3
The importance of variables.....	4
Rules for naming variables	4
A2. IF/ELSE STATEMENTS.....	4
“If” Statement	4
“If...else” Statements	4
Multiple conditions	5
A3. LOOPS.....	6
For loop	6
For loop in <i>range()</i>	7
While loop	8
PART B.....	9
Exercise 1.....	9
Flowchart	9
Python Input	9
Python Output.....	11
Exercise 2.....	12
Flowchart	12
Python Input	12
Python Output.....	13
Exercise 3.....	14
Flowchart	14
Python Input	15
Python Output.....	15
Exercise 4.....	17
Flowchart	17
Python Input	17
Python Output.....	18
Conclusion	20
Appendix.....	21
References.....	22

Introduction

In my assessment, I will cover something about software development I have learned at the college. The programming language I have learned is Python; to do so I am going to provide descriptions of some commands I am going to have to execute codes, and I'm going to show proof that in certain situations the software works positively, but also negatively.

Finally, I will solve my exercises, add codes and photos about the reaction of the programme.

Software Development

It is the process of developing computer software using one or more specific programming languages to address specific business or personal goals. Software development is typically a planned initiative that consists of a series of steps and stages that culminate in the creation of functional software.

As a result, a good definition would be "a process with a set of activities that create computer software products, including their design, development, testing, and implementation." The software is a collection of instructions or programmes that instruct a computer on what to do. It is hardware independent and allows computers to be programmed. [1]

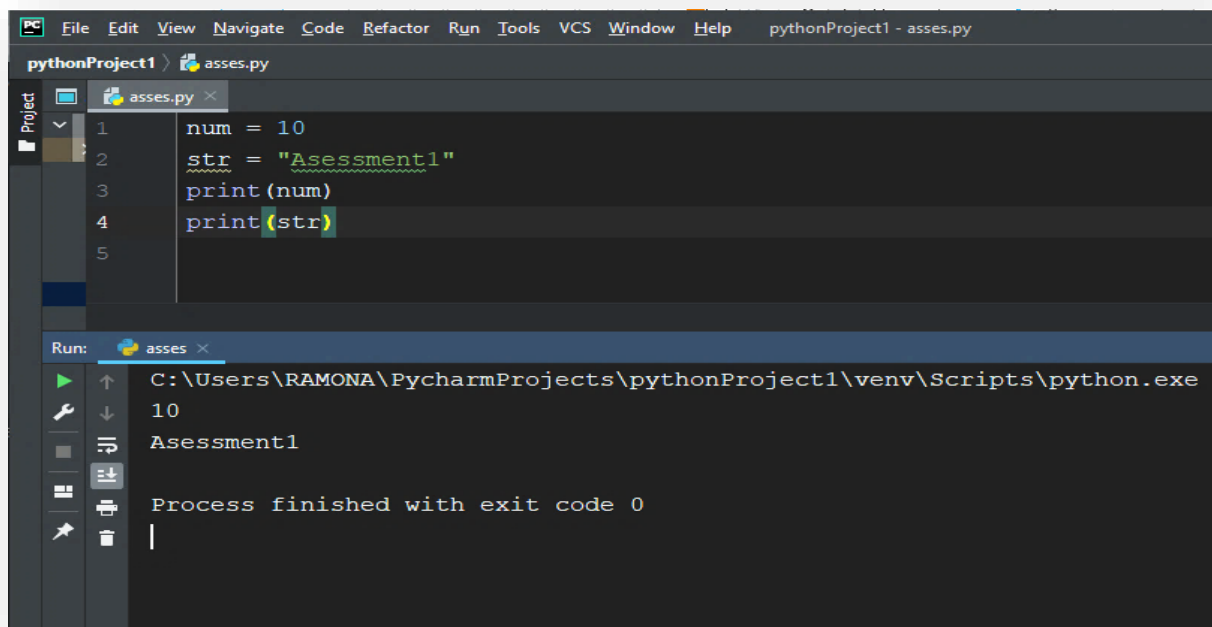
PART A

A1. VARIABLES

Definition

Variables are the beginning of all our studies. The use of variables, particularly in study, becomes the main root of everything. Variables were described as any entity that could assume various values, which means that any value other than the variables. [2]

- ✚ Numbers – integers (int)
- ✚ decimal numbers - float
- ✚ letters – string (str)



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python file named `asses.py` with the following code:

```
1 num = 10
2 str = "Aessment1"
3 print(num)
4 print(str)
5
```

Below the editor, the 'Run' console is visible, showing the execution of the script. The output is:

```
Run: asses
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe
10
Aessment1
Process finished with exit code 0
```

photo 1. Exemple

The importance of variables

Variables are important in computer programming because they allow me to write flexible programmes as a programmer. Rather than entering data directly into a programme, I can represent the data with variables. When the programme is run, the variables are replaced with actual data. [3]

Some programming languages require that we declare the form of our variable ahead of time so that the software can allocate memory for it. Strongly-typed is the expression for this.

Python was the programming language I studied at university. Python doesn't have a command for declaring variables. Instead, Python infers the type; however, there are a few things we can do to help define it.

Rules for naming variables

1. Variable names must begin with a letter or an underscore_, for example:
 - ❖ mySurname,
 - ❖ myCar,
 - ❖ myNo,
 - ❖ myAge.
2. Variable names are case-sensitive, for example:
 - ❖ myage, myAge and MYAGE are not the same.
3. We cannot use special characters (such as £\$%^&|~?!~)
4. A variable name can only contain alpha-numeric characters and underscore (A-Z, a-z, 0-9 and _). [4]

A2. IF/ELSE STATEMENTS

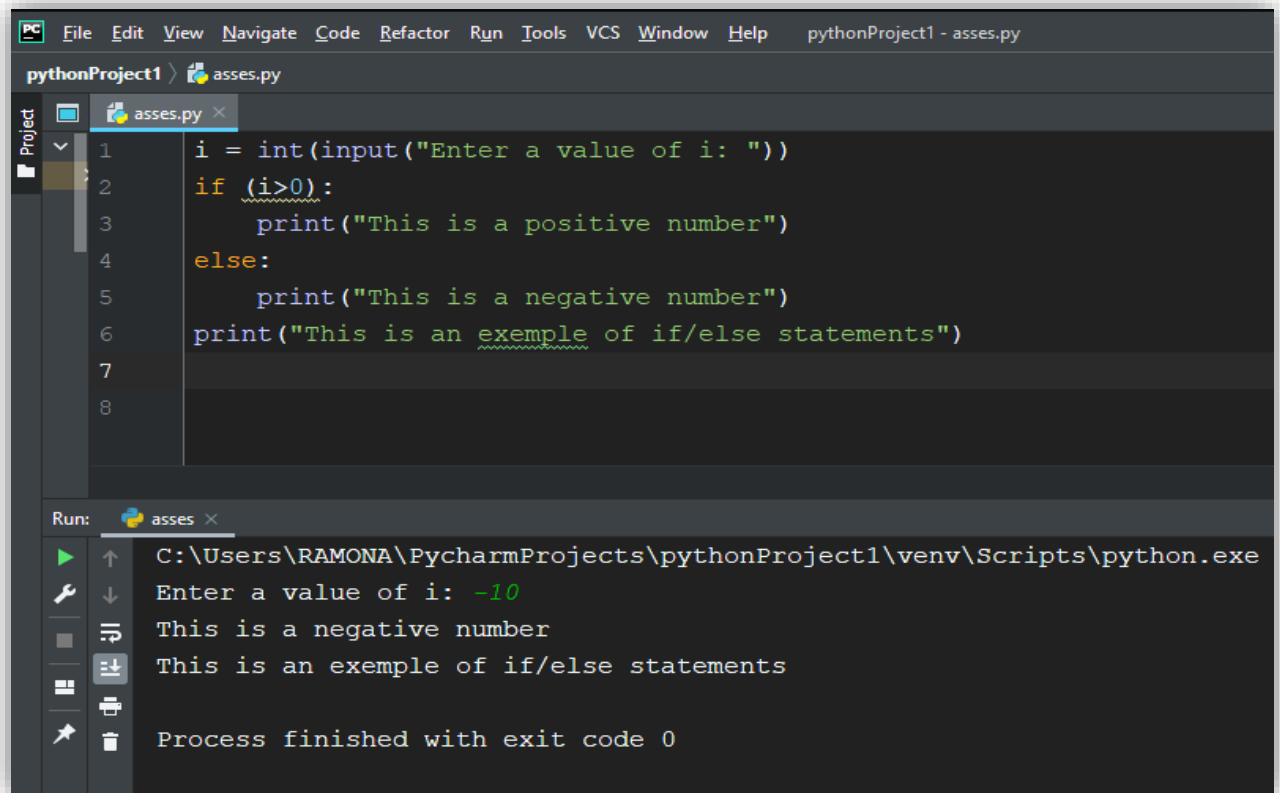
“If” Statement

If the instructions are flow controls that help me to execute a certain code only if a condition is met. I just want to print a message on the screen if the requirements are true, for example, then I can use the if statement in the programming process.

“If...else” Statements

If a particular condition is true, I use statements when I need to run some block of Python code. If..else statements are like an extension for 'if' statements, they can perform certain statements with the help of if.else, if the condition is true and if the condition is false, a different set of states. If the number is even and if the number is not even, I can do it with the if..else statement, for example, I would like to print a "even number." (*see photo 2*) [5]

- I use **if** to specify a block of code to be executed, if a specified condition is *true*.
- I use **else** to specify a block of code to be executed, if the same condition is *false*.



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project name is 'pythonProject1' and the file being edited is 'asses.py'. The code in the editor is as follows:

```
1 i = int(input("Enter a value of i: "))
2 if (i>0):
3     print("This is a positive number")
4 else:
5     print("This is a negative number")
6 print("This is an example of if/else statements")
7
8
```

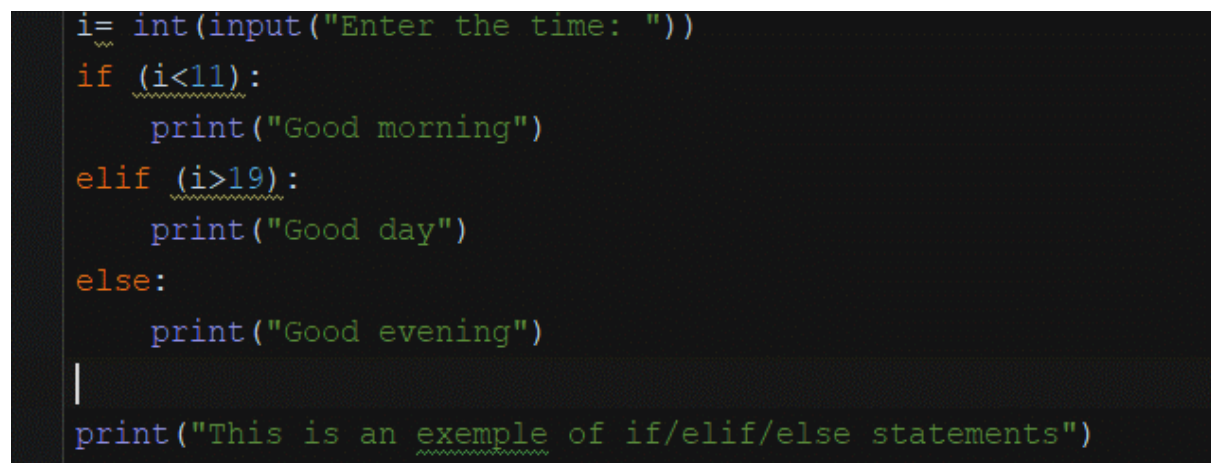
Below the editor, the 'Run' console shows the execution of 'asses.py'. The output is:

```
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe
Enter a value of i: -10
This is a negative number
This is an example of if/else statements
Process finished with exit code 0
```

photo 2.

Multiple conditions

The *if..elif..else statement* is used when I need to check multiple conditions. After *if* the condition, the *elif* conditions are applied. Python assesses the condition, and when *false* assesses it evaluates the *elif* blocks and executes the *elif* block which evaluates *true* expression. (see photo 3-input and photo 4 output)



The screenshot shows a code editor with the following Python code:

```
i= int(input("Enter the time: "))
if (i<11):
    print("Good morning")
elif (i>19):
    print("Good day")
else:
    print("Good evening")
|
print("This is an example of if/elif/else statements")
```

photo 3.

```
Enter the time: 10
Good morning

Enter the time: 14
Good evening

Enter the time: 21
Good day
This is an exemple of if/elif/else statements

Process finished with exit code 0
|
```

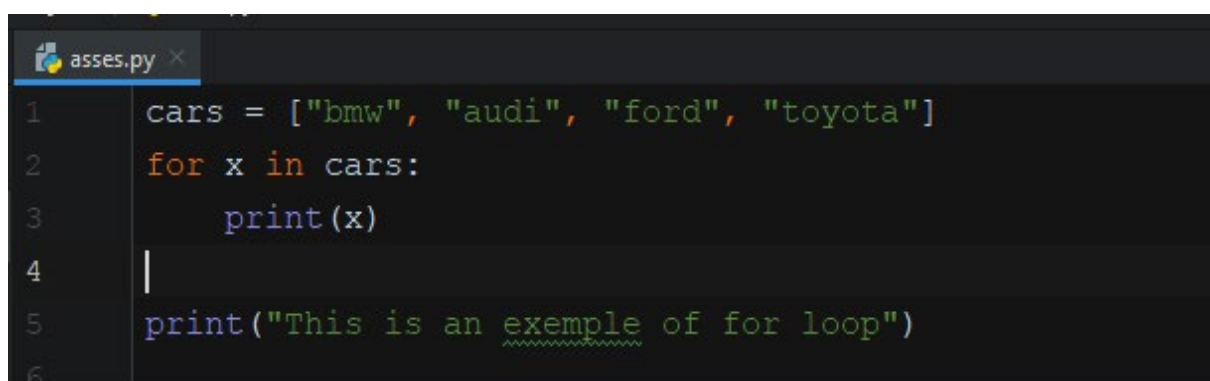
photo 4.

A3. LOOPS

Python offers three ways to perform loops, but I have learned only two of them which I will represent them here. Let's say I have to repeat some sections of the code often when I programme, these are known as loops. Loops have a number of iterations (how many times the loop is executed). In Python, I used two loop categories - *for* and *while*. Both exist in Python, but the accent for loop in Python is a bit different.

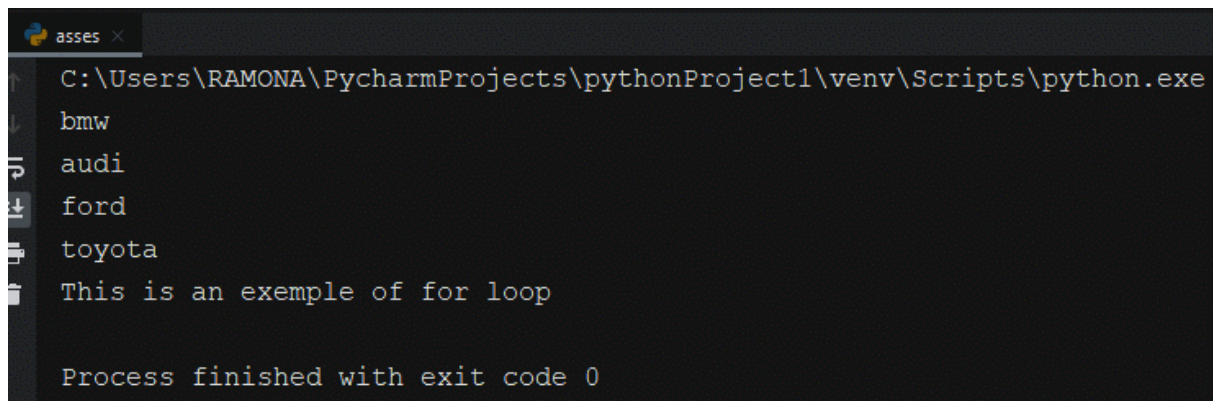
For loop

A *for loop* is used in sequence iteration (a list, a set or a string). This is less as in other languages and is more a way to iterate as in other objective programming languages. It works more like a method of iterator. The "for loop" enables us to execute several statements, once in a list for example for each item. (see photo 5-input and photo 6 output) [6]

A screenshot of a code editor window titled 'asses.py'. The code is as follows:

```
1 cars = ["bmw", "audi", "ford", "toyota"]
2 for x in cars:
3     print(x)
4
5 print("This is an exemple of for loop")
6
```

photo 5.



```

asses x
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe
bmw
audi
ford
toyota
This is an exemple of for loop

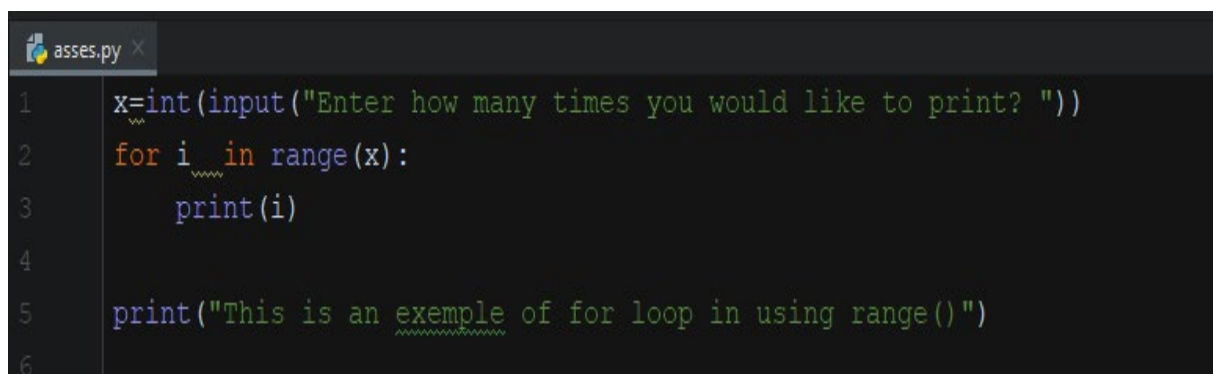
Process finished with exit code 0

```

*photo 6.**For loop in range()*

I can use the *range()* function to loop through a code set for a certain number of times. Range() returns a number sequence, beginning with 0 and increments with 1 by default, which ends with a specified number. I was iterating through a list using a loop in this example. I used a range() function to iterate by range-defined numbers for loops (). For example:

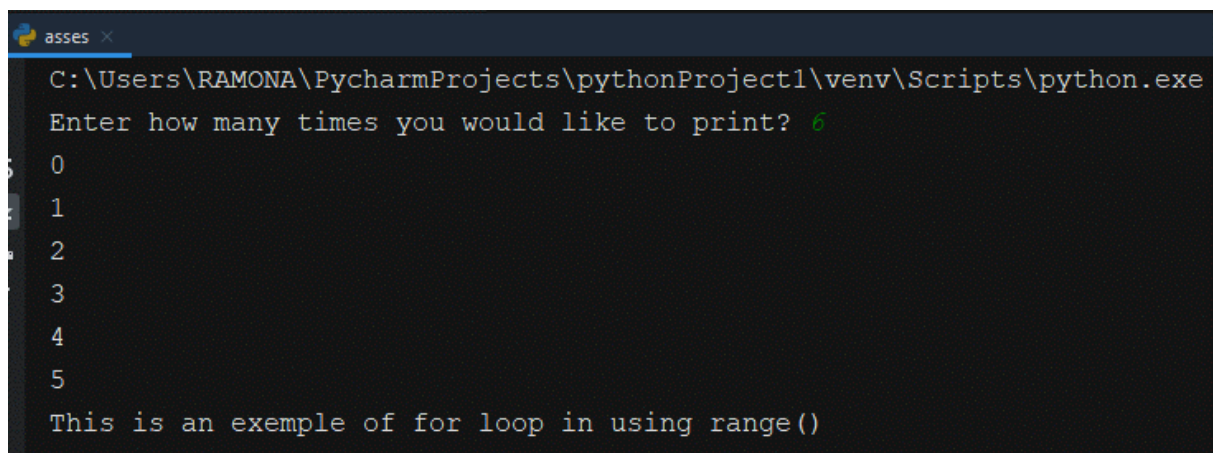
Range(x): produces a set of entire numbers from 0 to (x-1):



```

asses.py x
1  x=int(input("Enter how many times you would like to print? "))
2  for i in range(x):
3      print(i)
4
5  print("This is an exemple of for loop in using range()")
6

```

photo 7. Input.


```

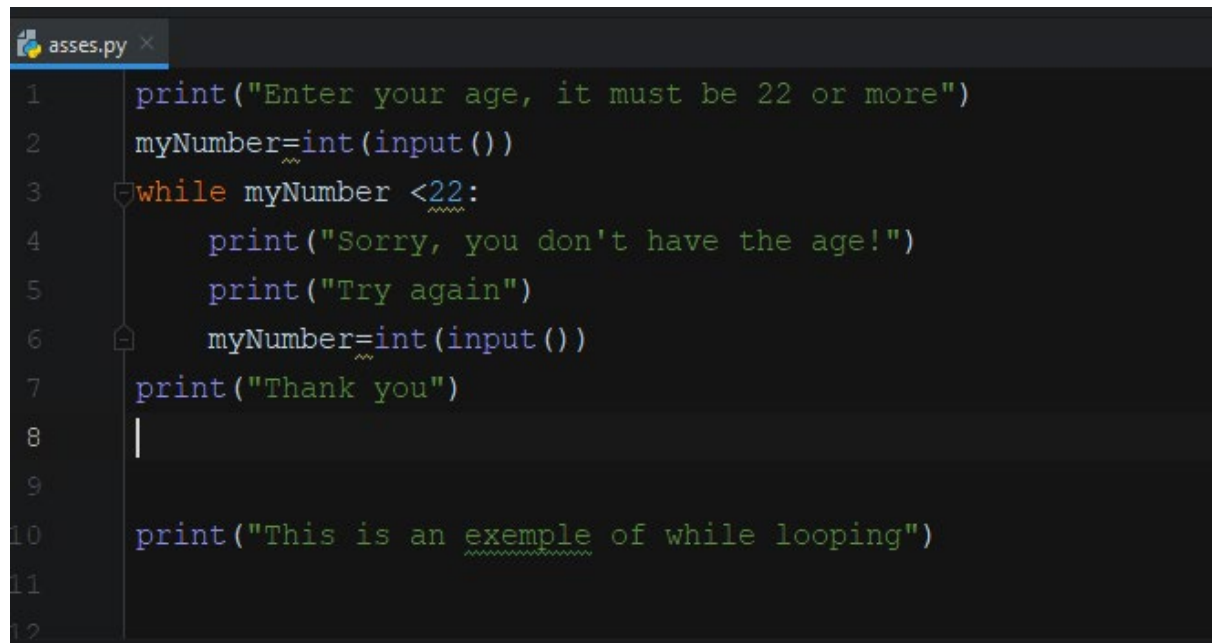
asses x
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe
Enter how many times you would like to print? 6
0
1
2
3
4
5
This is an exemple of for loop in using range()

```

photo 8. Output

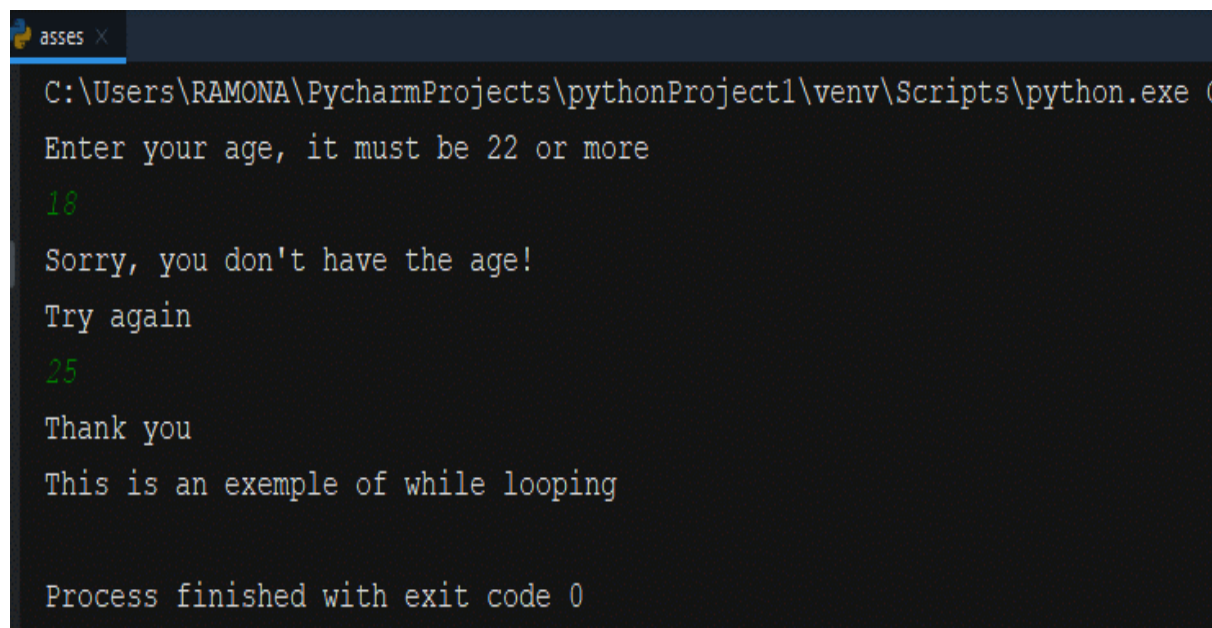
While loop

In python, *while loop* is used for programme a block of statements until the condition is met. And if the condition gets false, the line is executed immediately after the loop. (see *photo 9-input and photo 10 output*)



```
asses.py x
1 print("Enter your age, it must be 22 or more")
2 myNumber=int(input())
3 while myNumber <22:
4     print("Sorry, you don't have the age!")
5     print("Try again")
6     myNumber=int(input())
7 print("Thank you")
8 |
9
10 print("This is an exemple of while looping")
11
12
```

photo 9. Input.



```
asses x
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe C
Enter your age, it must be 22 or more
18
Sorry, you don't have the age!
Try again
25
Thank you
This is an exemple of while looping

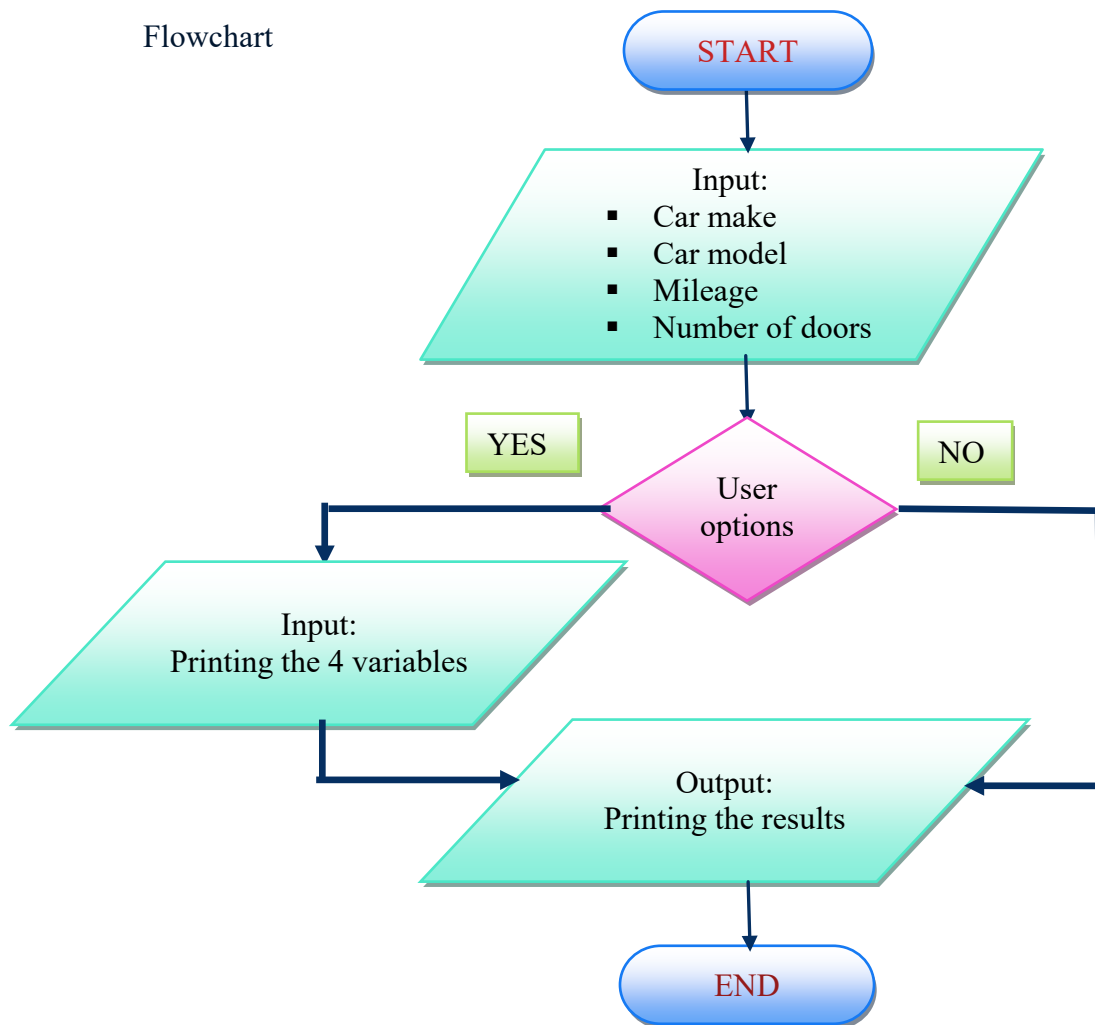
Process finished with exit code 0
```

photo 10. Output.

PART B

Exercise 1

Flowchart

*photo 11. Flowchart exercise 1.*

Python Input

The program I wrote allows the user to enter four variables. After the user enters these values, they will be displayed on the screen. I also introduced an option that allows the user to choose if he wants to enter once again these variables and the option to refuse.

I used:

1. *Two options:*

- *Option 1 "Yes":* Allows the user to re-enter the car details.
- *Option 2 "No":* If the user chooses this option he will not re-enter these options.

```

#Exercise 1
#The customer uses my programme to enter his car details.

#INPUT (The code seen by the programmer)
print("Please enter your car details! ")

#I added an extra command to give importance to the user:
userName = input("Please enter your name: ") #user enter his name
userName = userName.upper()

#The user enters the car details:
carMake = input("Insert car make: ") #user enter car make
carModel = input("Insert car model: ") #user enter car model
carMileage = input("Insert car mileage: ") #user enter car mileage
noDoors = input("Insert the number of car doors: ") #user enter the numbers
of the doors

print(sep='') #The "free space" command to differentiate what the user
enters and what he get

#OUTPUT (What the user will see after entering the car details)
print("Name: " + userName) #the program is printing the username
print("Car Make: " + carMake) #the program is printing the car make
print("Car Model: " + carModel) #the program is printing the car model
print("Car Mileage: " + carMileage) #the program is printing the car
mileage
print("Numbers of doors: " + noDoors) #the program is printing the car
mileage

print(sep='')

print("Do you want to try again? Please choose an option: ") #the user is
choosing an option
print("1. Yes") # 1 option
print("2. No") # 2 option

userOption = int(input())
if userOption == 1: #condition and first option for user
    carMake = input("Insert car make: ") # user enter car make
    carModel = input("Insert car model: ") # user enter car model
    carMileage = input("Insert car mileage: ") # user enter car mileage
    noDoors = input("Insert the number of car doors: ") # user enter the
numbers of the doors

    print( sep= '')

    print("Name: " + userName) # the program is printing the username
    print("Car Make: " + carMake) # the program is printing the car make
    print("Car Model: " + carModel) # the program is printing the car
model
    print("Car Mileage: " + carMileage) # the program is printing the car
mileage
    print("Numbers of doors: " + noDoors) # the program is printing the
car mileage
elif userOption == 2: #second option for user
    print("You choose to not continue! Anyway, great car!")
else: #option when 1 and 2 options are not avaiable
    print("Thank you!")

print("End of program!")

```

Python Output

Positive way: This picture shows that the program works correctly, and the user chooses the first option in which he can re-enter the car details.

```
C:\Users\RAMONA\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICE 1.py"
Please enter your car details!
Please enter your name: Ramona
Insert car make: Mercedes
Insert car model: C class
Insert car mileage: 145100
Insert the number of car doors: 5

Name: RAMONA
Car Make: Mercedes
Car Model: C class
Car Mileage: 145100
Numbers of doors: 5

Do you want to try again? Please choose an option:
1. Yes
2. No
1
Insert car make: BMW
Insert car model: X5
Insert car mileage: 98500
Insert the number of car doors: 3

Name: RAMONA
Car Make: BMW
Car Model: X5
Car Mileage: 98500
Numbers of doors: 3
End of program!
```

Negative way: This image in the program shows that the user chooses the second option being a negative value and will not allow the user to re-enter the car details.

```
C:\Users\RAMONA\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICE 1.py"
Please enter your car details!
Please enter your name: Ramona
Insert car make: Peugeot
Insert car model: Coupe
Insert car mileage: 75000
Insert the number of car doors: 3

Name: RAMONA
Car Make: Peugeot
Car Model: Coupe
Car Mileage: 75000
Numbers of doors: 3

Do you want to try again? Please choose an option:
1. Yes
2. No
2
You choose to not continue! Anyway, great car!
End of program!
```

Exercise 2

Flowchart

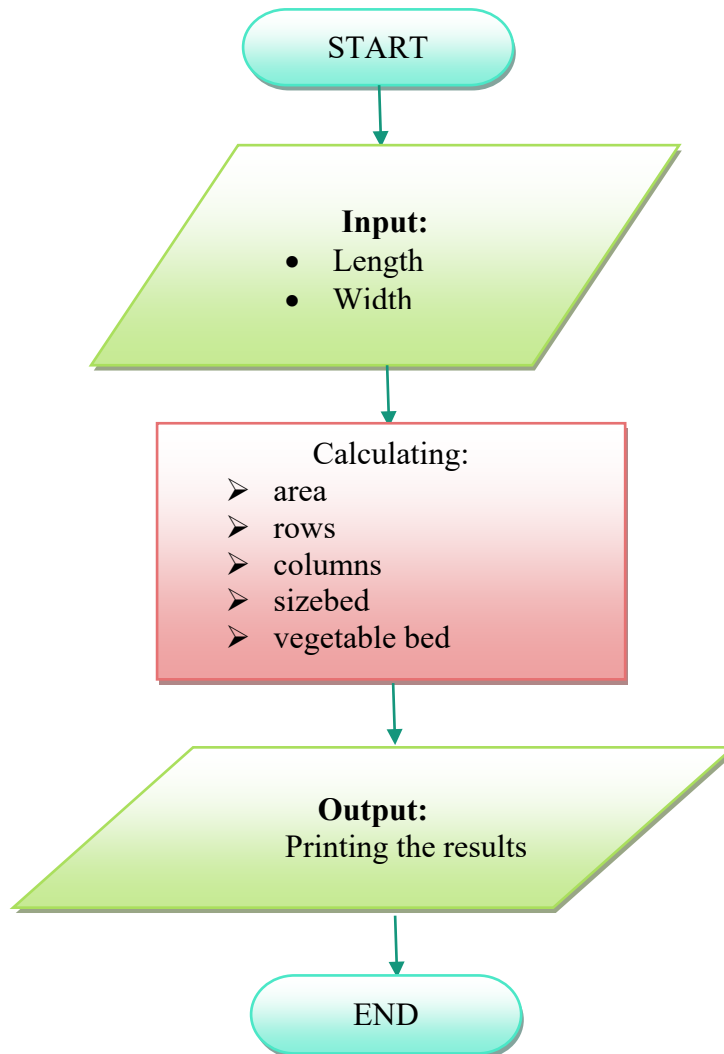


photo 12. Flowchart exercise 2.

Python Input

I have written this programme in order to enable a person who does not have experience in carrot growing, but only a minimum level of knowledge, to implement the dimensions of the bed of vegetables and to measure, secondly, the number of carrots that can grow in the bed.

The program allows the person to enter the length and width available to them to plant carrots. The program calculates the area according to the dimensions entered by it.

Among these, it will calculate how many rows and how many columns will be in the vegetable bed, calculating to the user how many whole carrots can be planted in that area.

```
#Exercice 2

print("Please enter the next details ") #the user is typing the next
details

print("Please enter the length of the vegetable bed: ") #the user is typing
the length of the land
bedLength = int(input()) #input for length

print("Please enter the width of the vegetable bed: ") #the user is typing
the width of the land
bedWidth = int(input()) #input for width

area = bedLength * bedWidth #the formula for calculating the area
print("The area of the bed vegetable is: ", area, "cm2") #the program
displays the area of the land

#Land dimensions
row = bedLength//4 #the formula for calculating the rows
column = bedWidth//4 #the formula for calculating the columns
print("Titles in row is: ", row) #the program displays how many rows will
be in the land
print("Titles in column is: ", column) #the program displays how many
columns will be in the land

sizeBed = row * column #the program displays how many rows and columns will
be in the land

print(sep= ' ')

vegetableBed = sizeBed//4 #the formula for calculating how many carrots can
grow in hsi vegetable bed
#the program displays how many carrots can grow in vegetable bed
print("In this bed you can plant: ", vegetableBed, "carrots")

print("End of program")
```

Python Output

Positive way: The program is printing for the user how many carrots can grow in the vegetable bed:

```
C:\Users\RAMONA\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICES 2.py"
Please enter the next details
Please enter the length of the vegetable bed:
24
Please enter the width of the vegetable bed:
31
The area of the bed vegetable is: 744 cm2
Titles in row is: 6
Titles in column is: 7

In this bed you can plant: 10 carrots
End of program
```

Negative mode: The program prints a negative response for the user when entering the length and width of the vegetable bed, but it does not meet the necessary conditions for growing at least 1 carrot:

```
C:\Users\RAMONA\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICES 2.py"
Please enter the next details
Please enter the length of the vegetable bed:
3
Please enter the width of the vegetable bed:
5
The area of the bed vegetable is: 15 cm2
Titles in row is: 0
Titles in column is: 1

In this bed you can plant: 0 carrots
End of program
```

Exercise 3

Flowchart

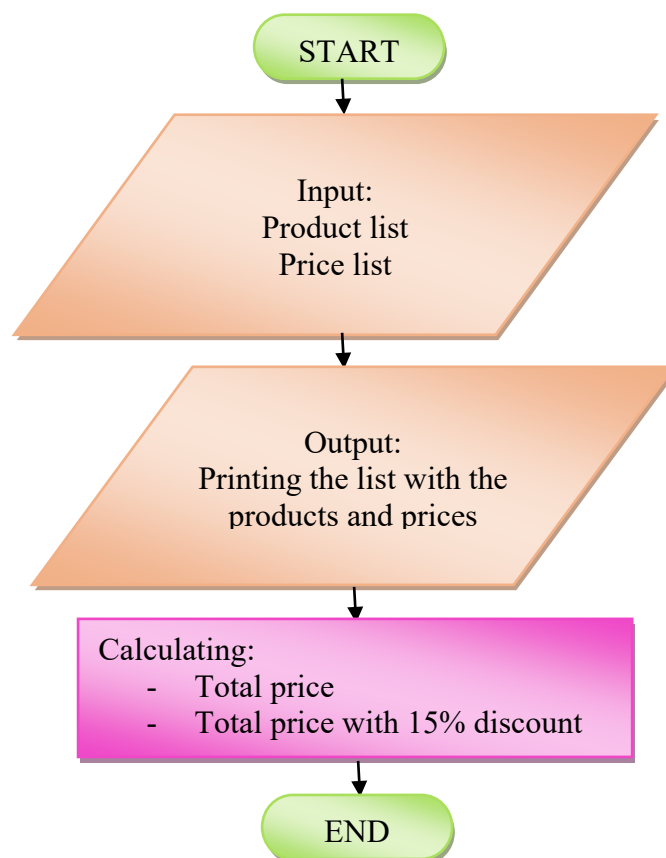


photo 13. Flowchart exercise 3.

Python Input

My program allows the user to enter the names of 5 products with prices, being displayed at the end in a list, their total is calculated according to the prices entered and then recalculated with a 15% discount.

I have introduced a condition, that if he has to write the price of the product, he can enter only numbers and not letters, allowing the program to calculate the total and the final price with added discount.

```
#Exercise 3

#condition
def c_input(i):
    while True:
        try:
            user_input = float(input(i))
        except ValueError:
            print("Please enter a valid number!")
            continue
        else:
            return user_input

#PRODUCT LIST AND PRICE LIST
productLst = [] #empty list
priceLst = [] #empty list

#user will introduce the next products and prices for the list
for m in range(1,6):
    print("Product", m, ":")
    product = input(" ") #user is adding the products in the list
    print("Price", m, ":")
    price = c_input(" ") #user is adding the prices in the list
    print(sep= ' ') #break in the list
    productLst.append(product)
    priceLst.append(price)

#the program is calculating the total price of the products
totalPrice = sum(priceLst) #formula for total price
print("List of the products: ") #display the list
print()
for m in range(0,len(productLst)):
    print(productLst[m], " ", priceLst[m]) #display the lists
print("Total price of the products: ", totalPrice) #displaying the total price

#the program is calculating the total price with 15% discount
d = totalPrice - ((totalPrice*15)/ 100) #formula for calculating the discount
print("Price with discount: ", d ) #displaying the price with discount

print("End of program")
```

Python Output

Positive way: The program is printing for the user the list with products and prices. Is calculating the total price and the price with 15% discount:

```

EXERCICE 3
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe "C:/Users/RAMONA/PycharmProjects/pythonProject1/EXERCICE 3.py"
Product 1 :
Cabbage
Price 1 :
4

Product 2 :
Broccoli
Price 2 :
5.50

Product 3 :
Onion
Price 3 :
3.50

Product 4 :
Corn
Price 4 :
5

Product 5 :
Cucumber
Price 5 :
2.50

List of the products:

Cabbage 4.0
Broccoli 5.5
Onion 3.5
Corn 5.0
Cucumber 2.5
Total price of the products: 20.5
Price with discount: 17.425
End of program

```

Negative mode: The program prints a negative response for the user when is entering letters on the price, instead of numbers and gives the chance for user to insert again numbers:

```

EXERCICE 3 (1)
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICE 3.py"
Product 1 :
cabbage
Price 1 :
tomato
Please enter a valid number!
5.12

Product 2 :
tomato
Price 2 :
6.54

Product 3 :
cucumber
Price 3 :
4.15#
Please enter a valid number!
4.15

Product 4 :
carrot
Price 4 :
1.45

Product 5 :
onion
Price 5 :
2.13

List of the products:

cabbage 5.12
tomato 6.54
cucumber 4.15
carrot 1.45
onion 2.13
Total price of the products: 19.39
Price with discount: 16.4815
End of program

```

Exercise 4

Flowchart

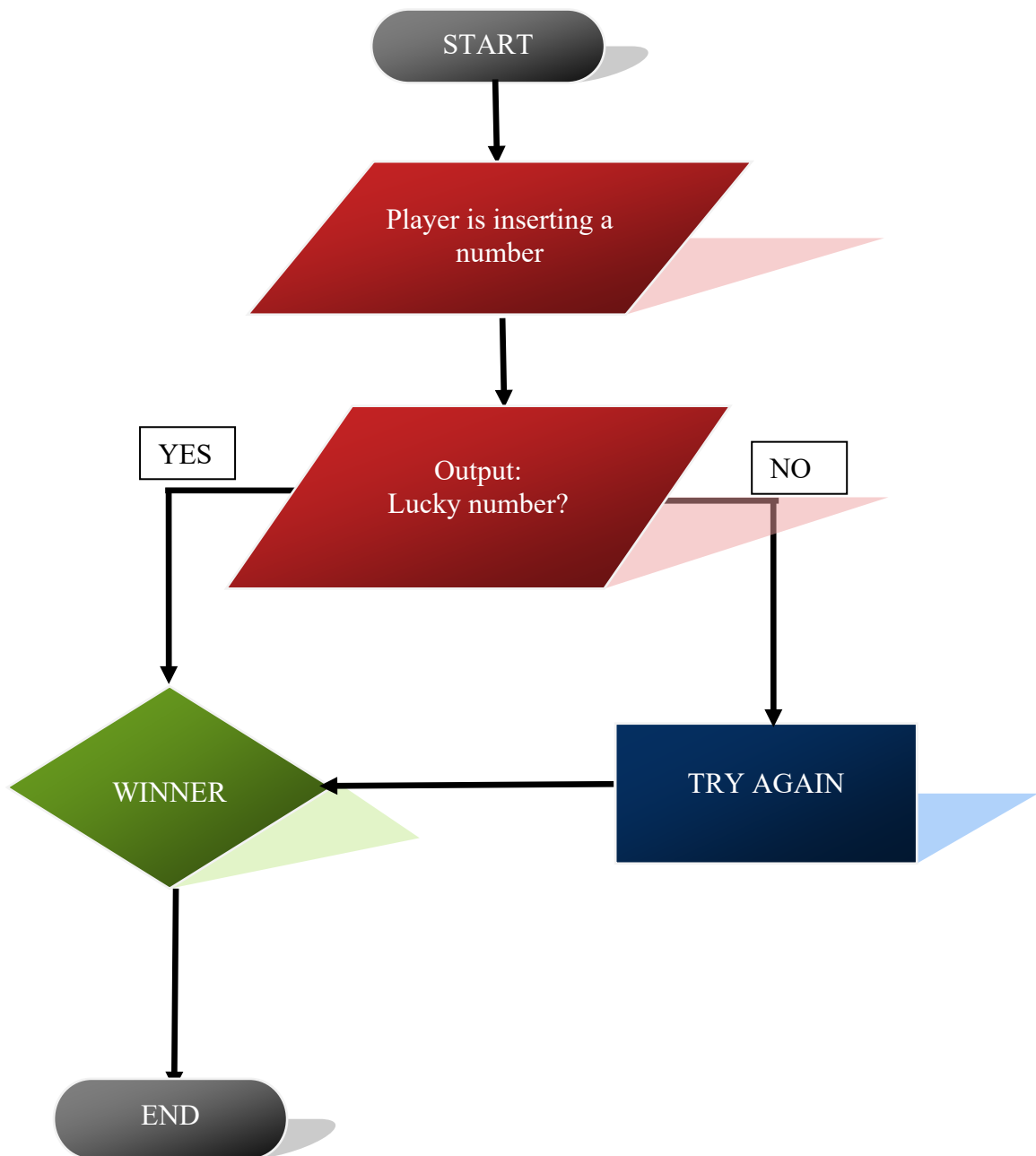


photo 14. Flowchart exercise 4.

Python Input

My program is a treasure hunt, the user enters a number from 0 to 15 looking for a treasure, he can enter a number 2000 times, one of them being the winner:

```
#Exercise 4 Treasure Hunt

luckyList = ["0 ", "0 ", "0 ", "0 ", "X "]
n=0 #how many times the user is trying to guess the number
while n!=2000:
    a=[]
    b=[]
    print("Choose position: (0-15)")
    user = int(input()) #user is inserting the number
    import random
    for r in range(0,4):
        z1 = random.choice(luckyList)
        z2 = random.choice(luckyList)
        z3 = random.choice(luckyList)
        z4 = random.choice(luckyList)
        b.append(z1)
        b.append(z2)
        b.append(z3)
        b.append(z4)
        c = [z1,z2,z3,z4]
        a.append(c)
#Condition
    if b[user]=="X ":
        print("Win! Well done. You took ",n," turns")
        print(sep= ' ')
    else:
        print("Miss! Unlucky")
        print("".join(a[0]))
        print("".join(a[1]))
        print("".join(a[2]))
        print("".join(a[3]))
        n = n + 1
```

Python Output

Positive way: The program displays a message for the player if he wrote the lucky number and won the treasure hunt:

```
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICE 4.py"
Choose position: (0-15)
5
Miss! Unlucky
X X X 0
0 0 X 0
0 X 0 0
0 0 X 0
Choose position: (0-15)
4
Win! Well done. You took 1 turns
```

Negative mode: The program allows the player to enter a number until that number becomes the winner. Many times, he can only guess that number by trying several times:

```

EXERCICE 4 (1) x
C:\Users\RAMONA\PycharmProjects\pythonProject1\venv\Scripts\python.exe "C:/Users/RAMONA/Desktop/ASSESSM PYON/EXERCICE 4.py"
Choose position: (0-15)
5
Miss! Unlucky
0 0 0 0
0 0 0 0
X 0 0 0
X 0 0 0
Choose position: (0-15)
9
Miss! Unlucky
0 X X 0
0 0 0 X
0 0 0 X
0 0 X 0
Choose position: (0-15)
4
Miss! Unlucky
0 X X 0
0 0 X 0
0 0 0 0
0 0 0 X
Choose position: (0-15)
7
Miss! Unlucky
0 0 0 0
0 0 0 0
0 X 0 0
0 X 0 0
Choose position: (0-15)
3
Win! Well done. You took 4 turns

```

Conclusion

Finally, I studied part of the development of software firstly I learned the language of programming: Python, I applied learning commands, the conditions that must be respected in certain situations and secondly, I showed that my knowledge is average, so that I can solve some exercises more easily, and some harder, from what I learned in the lessons and in those I studied.

Appendix



ASSESSM PYTHON EXERCICES.zip

References

- [1] IT Chronicles Media Inc., “IT Chronicles,” 2021. [Online].
Available: <https://itchronicles.com/what-is-software-development/>.
[Accessed 10 April 2021].
- [2] E. E. Onjining, “Academia,” 2021. [Online].
Available: <https://www.academia.edu/8420795/Variables>.
[Accessed 13 April 2021].
- [3] E. Linde, “Variables, Constants, and Data Types,” April 2015. [Online].
Available: <https://www.kidscodecs.com/variables-constants-data-types/>.
[Accessed 14 April 2021].
- [4] Pearson Education, Informit., “InformIT,” 6 July 2001. [Online].
Available: <https://www.informit.com/articles/article.aspx?p=131025&seqNum=3>.
[Accessed 16 April 2021].
- [5] TutorialsTeacher, “TutorialsTeacher,” 2020. [Online].
Available: <https://www.tutorialsteacher.com/python/python-if-elif>.
[Accessed 19 April 2021].
- [6] GeeksforGeeks, “GeeksforGeeks,” 12 November 2020. [Online].
Available: <https://www.geeksforgeeks.org/loops-in-python/>.
[Accessed 21 April 2021].