

**EXPERIMENT #2(A0**

**INTRODUCTION**

**The purpose of this lab is to get familiar with Advance concepts in Python**

|  |  |
| --- | --- |
| **Name** | Hadeer-Ur-Rehman |
| **Date** | 25/7/2020 |
| **Registration No** | 12413 |
| **Department** | BSCS |
| **Total Marks** |  |
| **Marks Obtained** |  |
| **Remarks** |  |
|  |
|  |

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Lab Instructor Signature**

**Experiment 2**

**INTRODUCTION**

**OBJECTIVE**

The goal of this tutorial is to understand advance topic of python

Students will be able to understand

* Conditional Statements: If,Else,Elif Statement In Python
* Loops In Python (For Loop ,While Loop Nested Loops In Python

**THEORY :**

**CONTROL STRUCTURES IF ELSE ELIF:**

An else statement can be combined with an if statement. An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.

The else statement is an optional statement and there could be at most only one else statement following if.

### **Syntax**

The syntax of the if...else statement is −

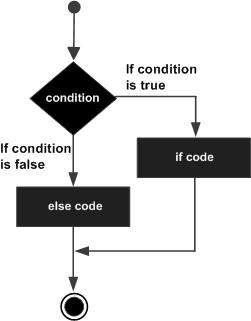
if expression:

statement(s)

else:

statement(s)

**Flow Diagram:**



**The elif Statement**

The elif statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.

Similar to the else, the elif statement is optional. However, unlike else, for which there can be at most one statement, there can be an arbitrary number of elif statements following an if.

**syntax**

if expression1:

statement(s)

elif expression2:

statement(s)

elif expression3:

statement(s)

else:

statement(s)

Core Python does not provide switch or case statements as in other languages, but we can use if..elif...statements to simulate switch case as follows −

**Demo Code:**

#!/usr/bin/python

var = 100

if var == 200:

print "1 - Got a true expression value"

print var

elif var == 150:

print "2 - Got a true expression value"

print var

elif var == 100:

print "3 - Got a true expression value"

print var

else:

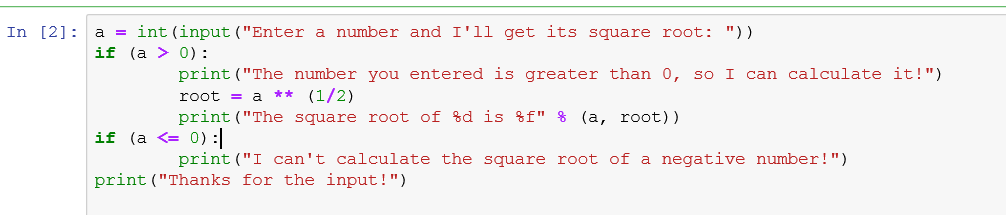
print "4 - Got a false expression value"

print var

print "Good bye!"

**Example: 01**

Print square root of negative or positive number using if and operators.



While we are looking at operators, there are another set used in writing conditions, these are called logical operators. There is an example of a logical operator.

if country == "England" or country == "england":

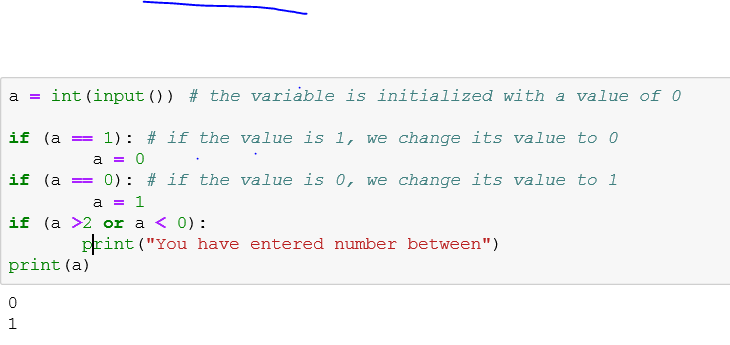
The logical operator used here is the word or. The interpreter evaluates the expressions either side of the **or** operator and if one or the other, or both expressions are true the result is true. There are several logical operators you can use in Python: **and, or, not.**

Conditions can be composed using two basic logical operators:

|  |  |
| --- | --- |
| **Operator** | **Syntax** |
| Logical AND | And |
| Logical OR | Or |

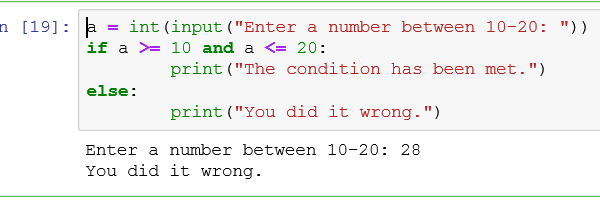
**Example: 02**

Write conditional statements to print value of 0 to 1 and 1 to 0 and numbers in between.



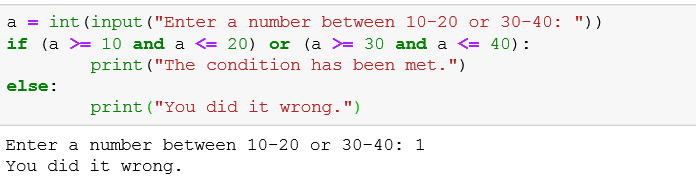
**Example: 03**

Let's look at another example:



**Example: 04**

Of course, operators can also be combined using parentheses:



**CONTROL STRUCTURES LOOP**

In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. There may be a situation when you need to execute a block of code several number of times.

Programming languages provide various control structures that allow for more complicated execution paths.

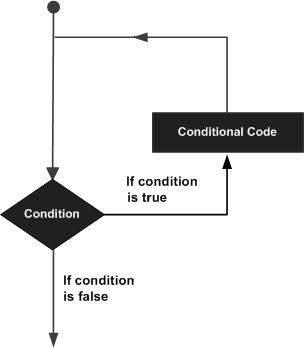
A loop statement allows us to execute a statement or group of statements multiple times.

**For Loop:**

**syntax:**

for iterating\_var in sequence:

statements(s)



**While - Loop**

Syntax:

**1initialization**

**2while (Condition):**

#body of the loop/statement goes here

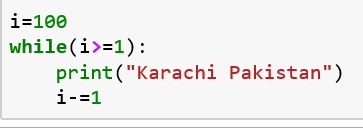
**3increment/decrement**

**Example #1**

Print Karachi Pakistan 100 times in a separate line

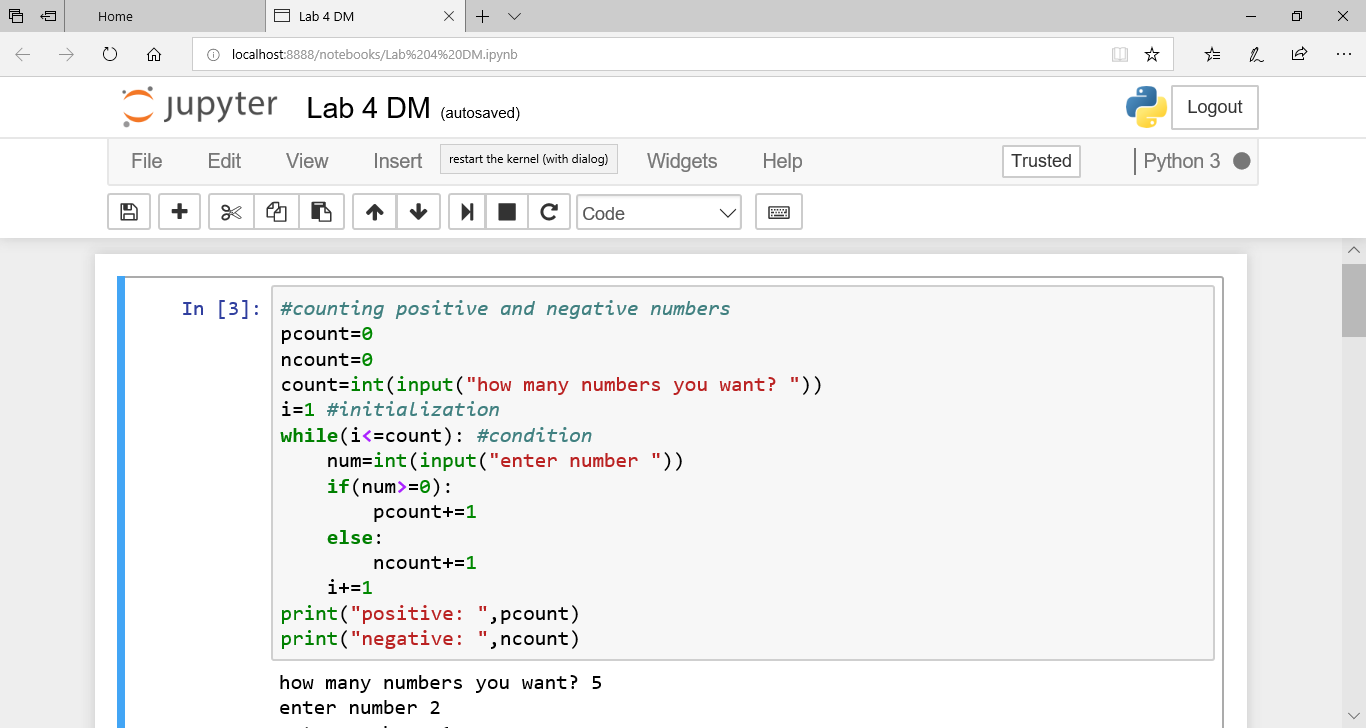


Another way to solve this problem with decrement operator



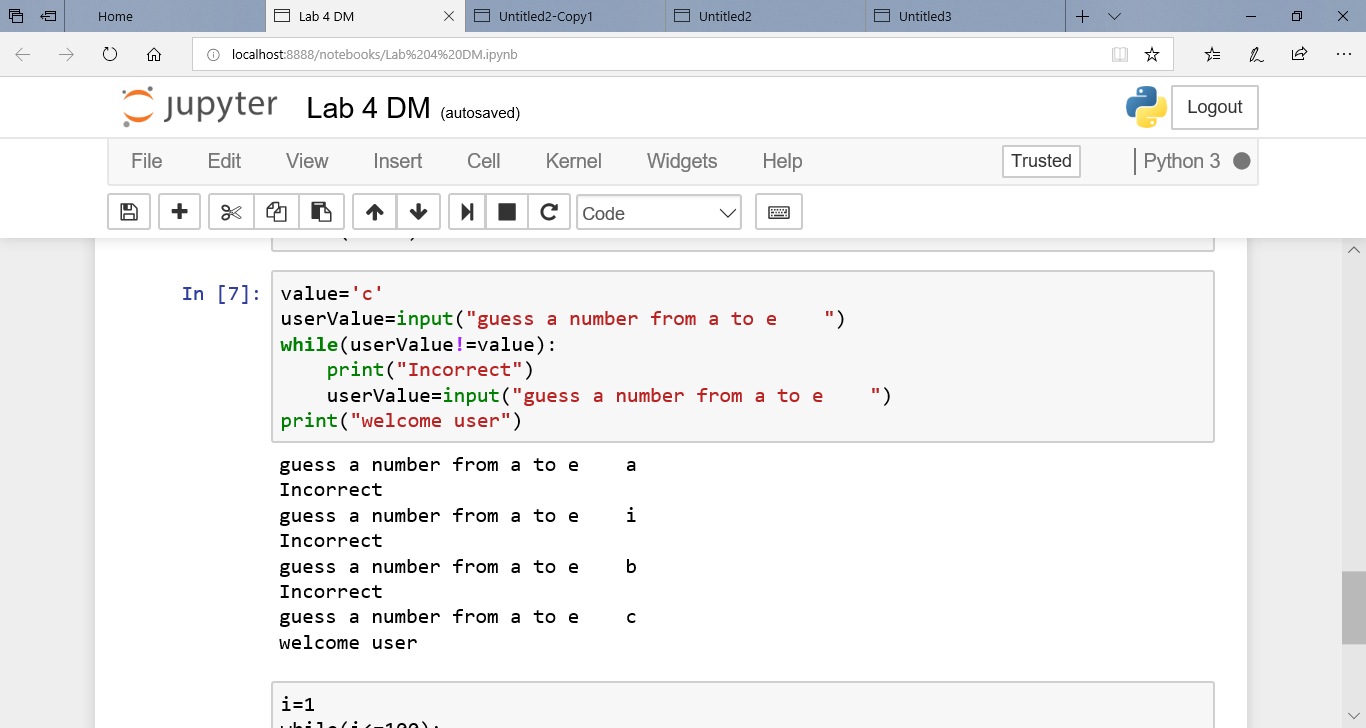
**Example # 2**

Take collection of number input from user. Print the total positive and negative number.



**Example # 3**

Fixed a Letter from a to e and then ask the user to guess that letter until correct letter entered.



**LOOP TYPE AND DESCRIPTION**

|  |  |
| --- | --- |
| **Sr.No.** | **Loop Type & Description** |
| 1 | [while loop](https://www.tutorialspoint.com/python/python_while_loop.htm)  Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body. |
| 2 | [for loop](https://www.tutorialspoint.com/python/python_for_loop.htm)  Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. |
| 3 | [nested loops](https://www.tutorialspoint.com/python/python_nested_loops.htm)  You can use one or more loop inside any another while, for or do..while loop. |

**LAB EXERCISES**

**Exercise 1:**

(I)Cabinets and Boxes are objects that are mostly in cubic shape. Make a program that takes inputs like height, width and depth from user and then calculate volume of the cube:

volume = height ∗ width ∗ depth

After calculating volume of cube, compare it with following ranges and print the relevant label:

|  |  |
| --- | --- |
| Volume Range | Label |
| 1 cm3 – 10 cm3 | Extra Small |
| 11 cm3 – 25 cm3 | Small |
| 26 cm3 – 75 cm3 | Medium |
| 76 cm3 – 100 cm3 | Large |
| 101 cm3 – 250 cm3 | Extra Large |
| 251 cm3 and above | Extra-Extra Large |
|  |  |

**Source Code:**

h=float(input("Enter Height:"))

w=float(input("Enter Width:"))

d=float(input("Enter Depth:"))

v=h\*w\*d

print("The Volume Of Cube is:",v)

if v >= 251:

print("Extra Extra Large")

elif v >= 101 and v <= 250:

print("Extra Large")

elif v >= 76 and v <= 100:

print("Large")

elif v >= 26 and v <= 75:

print("Medium")

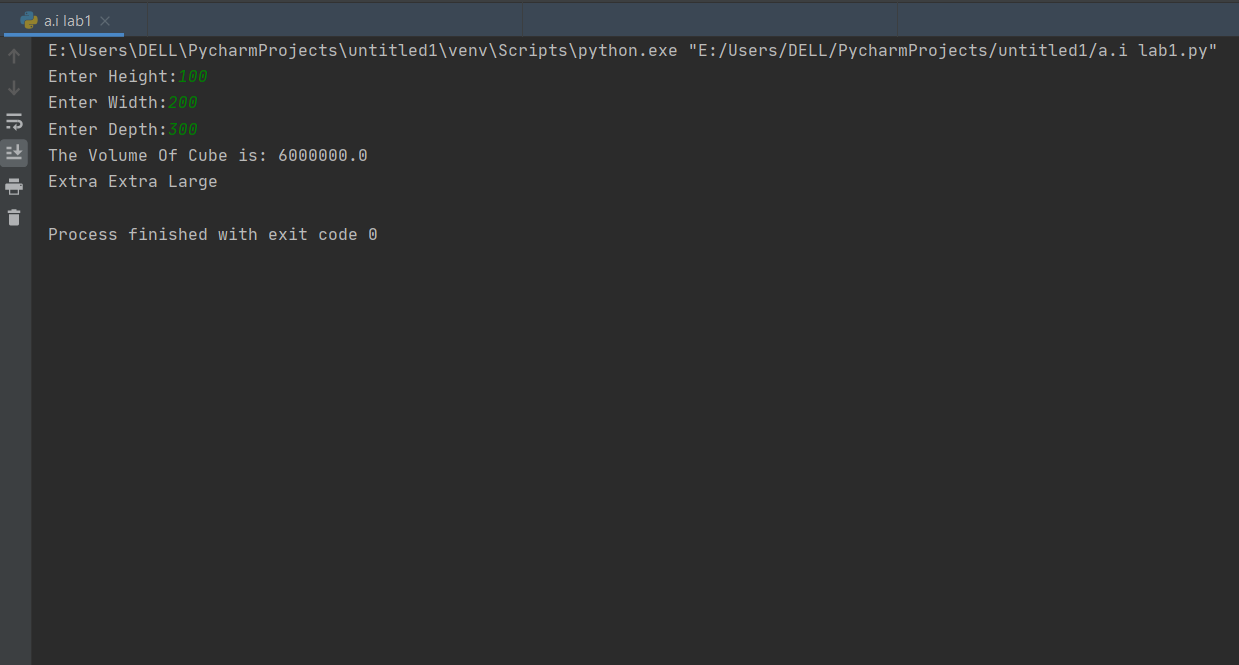
elif v >= 11 and v <= 25:

print("Small")

elif v >= 1 and v <= 10:

print("Extra Small")

**Output:**



(II)In a company ,worker efficiency is determined on the basis of the time required for a worker to complete a particular job.If the time taken by the worker is between 2-3 hours then the worker is said to be highly efficient. If the time required by the worker is between 3-4hours,then the worker is ordered to improve speed. If the time taken is between 4-5 hours ,the worker is given training to improve his speed ,and if the time taken by the worker is more than 5 hours ,then the worker haas to leave the company, If the time taken by the worker is input through the keyboard,find the efficiency of the worker.

**Source Code:**

time=float(input("Enter the time taken by the worker in hours (hh.mm) :- "))

if time>=2.00 and time<=3.00:

print("This worker is highly efficient.");

elif time>3.00 and time<=4.00:

print("This worker is ordered to improve speed.")

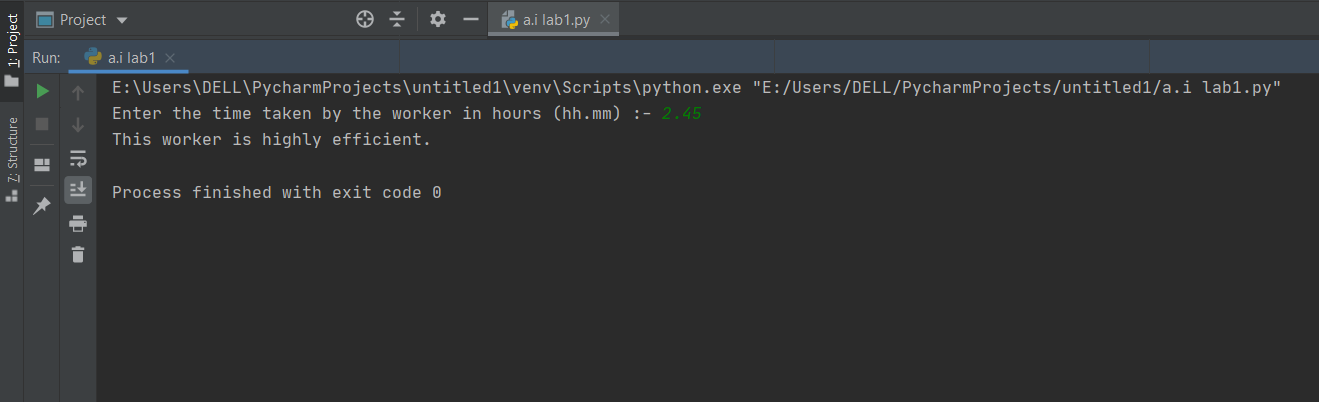
elif time>4.00 and time<=5.00:

print("This worker is given training to improve his speed. ")

else:

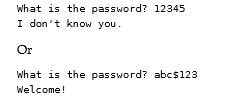
print("This worker has to leave the company.")

**Output:**



(iii)The program must prompt the user for a username and password. The program should compare the password given by the user to a known password. If the password matches, the program should display “Welcome!” If it doesn’t match, the program should display “I don’t know you.

*Note: the password should not be case sensitive and it’s value is abc$123 or ABC$123*



What is the password? ABC$123

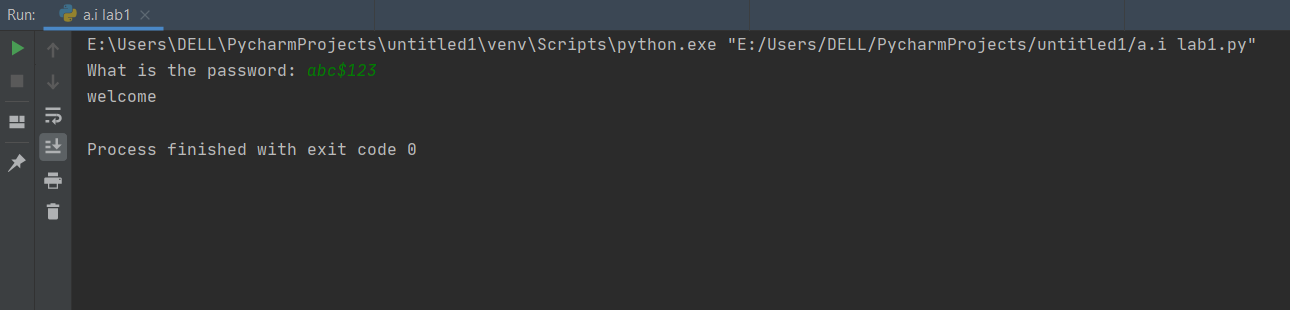
Welcome!

**Source Code:**

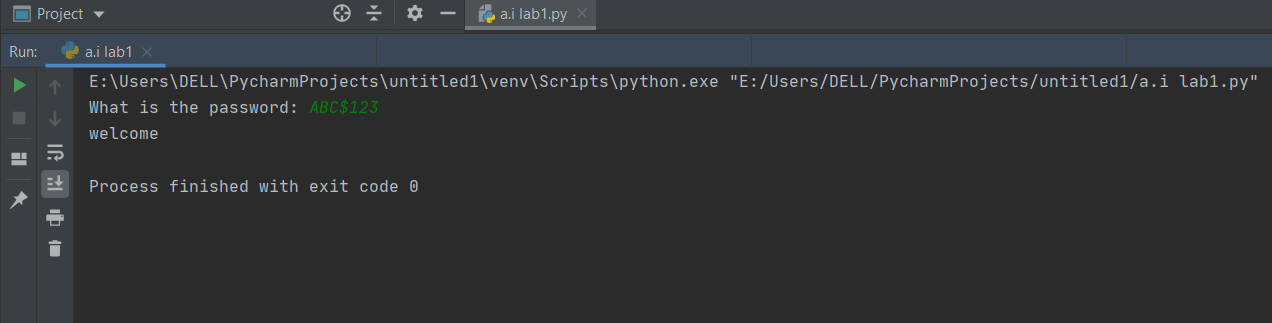
password = input("What is the password: ")  
password=password.lower()  
if password=='abc$123':  
 print("welcome")  
else:  
 print("i dont know you")

**Output:**

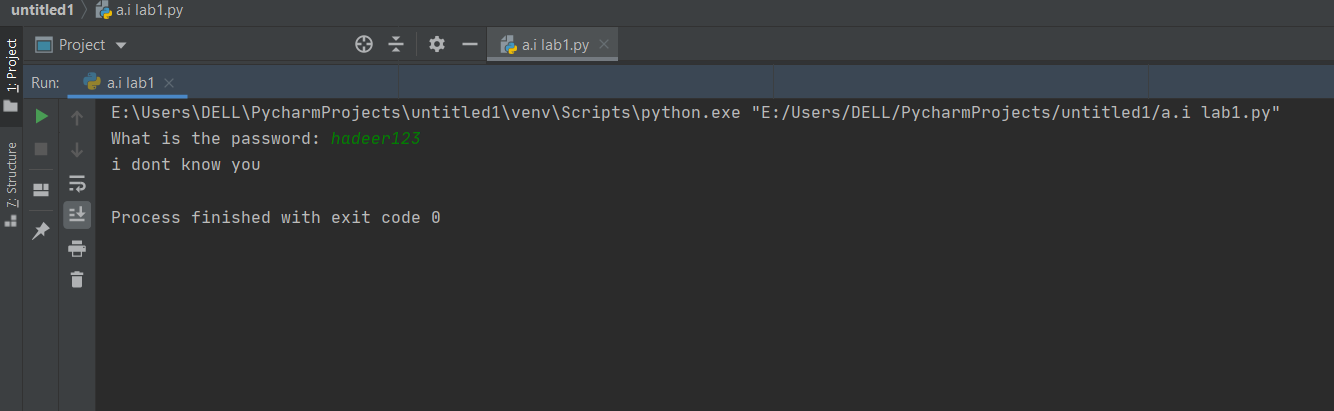
**In lower case:**



**In Uppercase:**



**Invalid Password:**



**Exercise 2:**

### (i)What Would Python Print?

>>> n = 3

>>> while n >= 0:

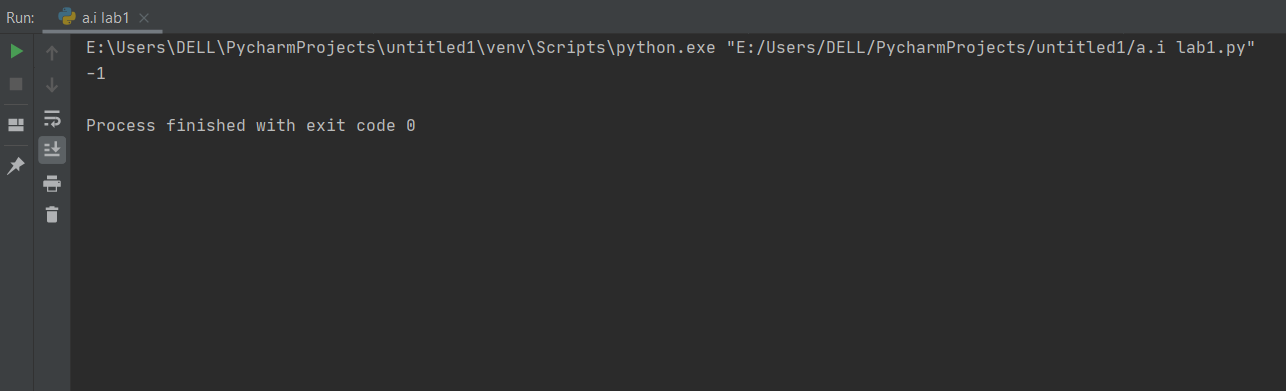
... n -= 1

... print(n)

...

\_\_\_\_\_\_

**Output:**



The code block will continue to run until n becomes < 0, since 0 is not greater than or equal to 0.

### (ii): What Would Python Print?

>>> # typing Ctrl-C will stop infinite loops

>>> n = 4

>>> while n > 0:

... n += 1

... print(n)

...

\_\_\_\_\_\_

Make sure your while loop condition eventually becomes false, or it'll never stop!

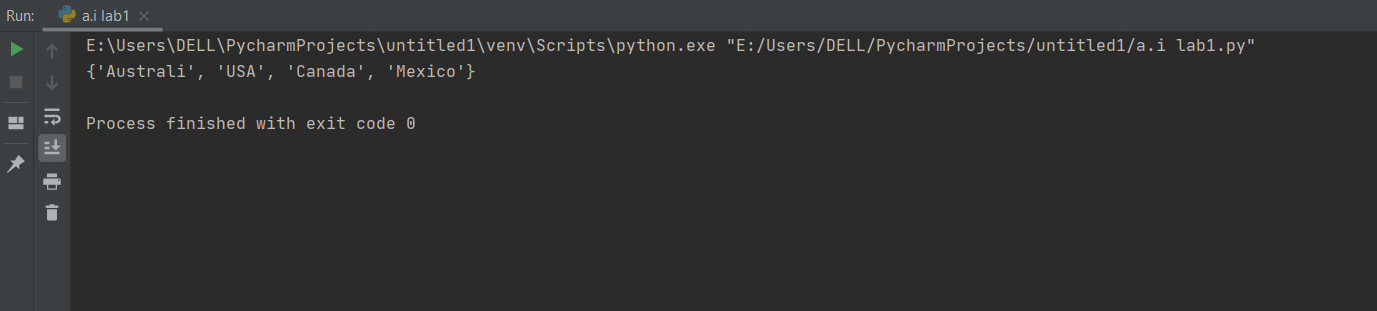
**Output:**



(ii)Try the cenario below:

Make a program that lists the countries in the set

|  |
| --- |
| clist=[‘Canada’,’USA’,’Mexico’,’Australi’] |
| Source Code:  clist=["Canada","USA","Mexico","Austral”]  s=set(clist)  print(s)  **Output:** |

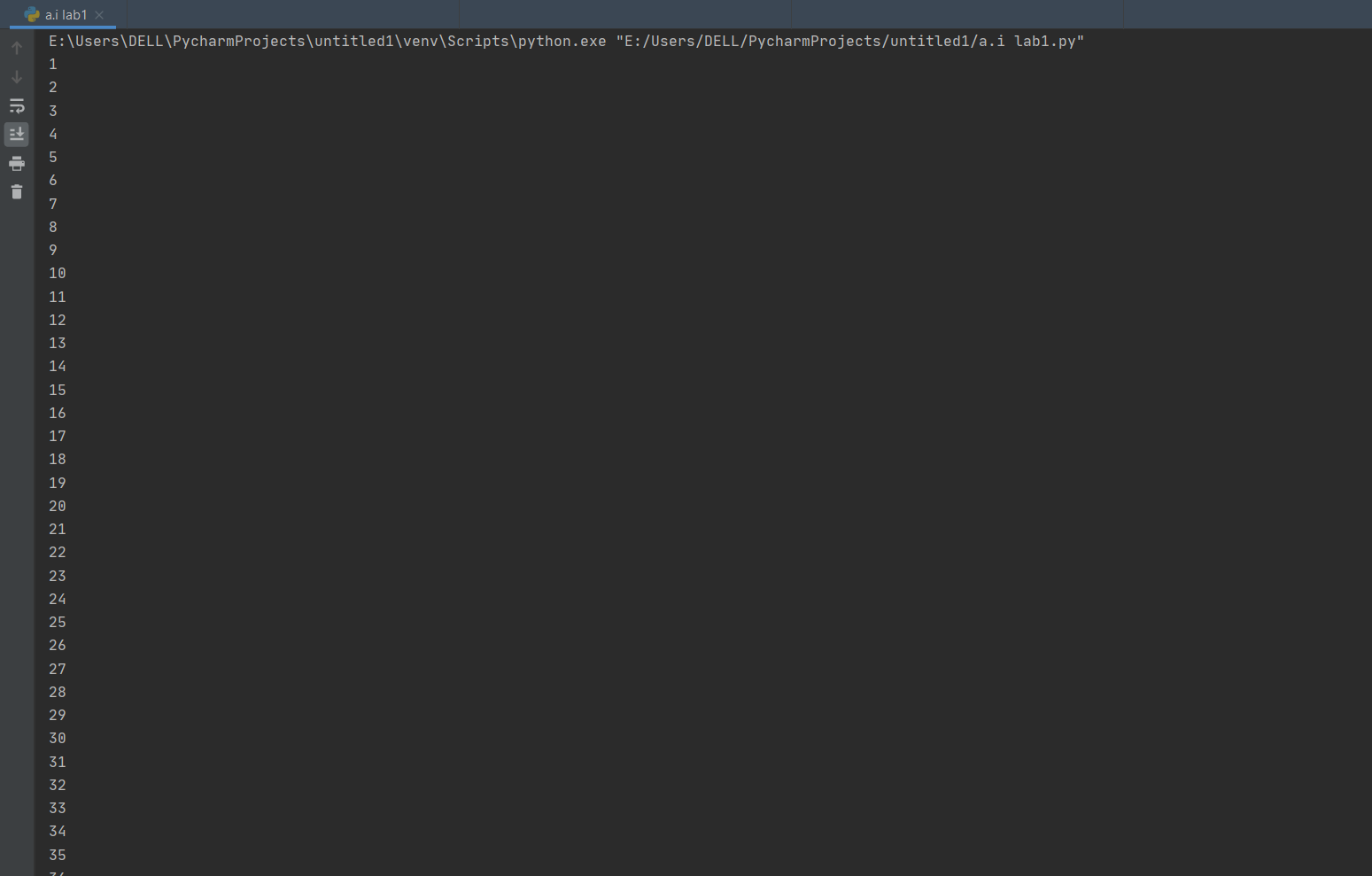


1. Create a loop that counts from 0 to 100

**Source Code:**

**number=1  
while number<=100:  
 print(number)  
 number=number+1**

**Output:**

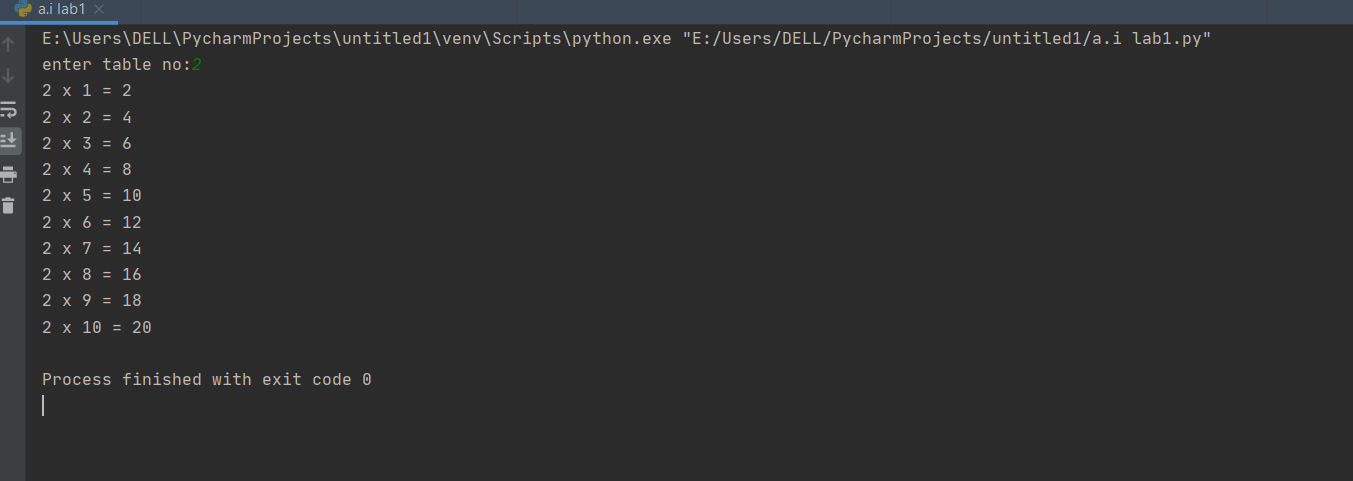


2. Make a multiplication table using a loop

**Source Code:**

tn=int(input("enter table no:"))  
for t in range(1,10):  
 print(tn, "x" , t , "=", tn\*t)

**Output:**

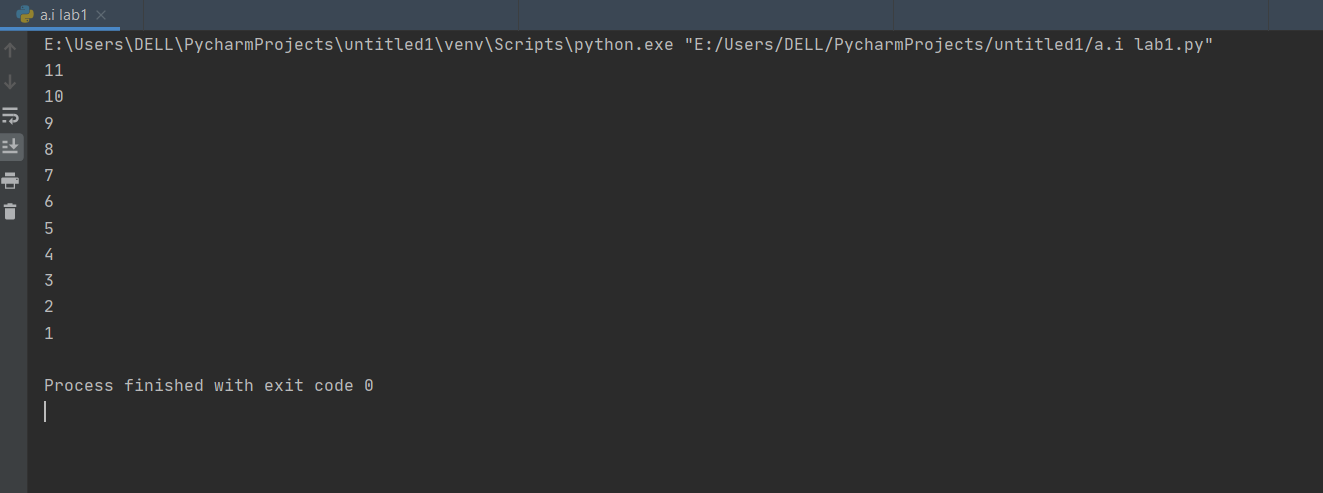


3. Output the numbers 1 to 10 backwards using a loop

**Source Code:**

for i in range(11,0,-1):  
 print(i)

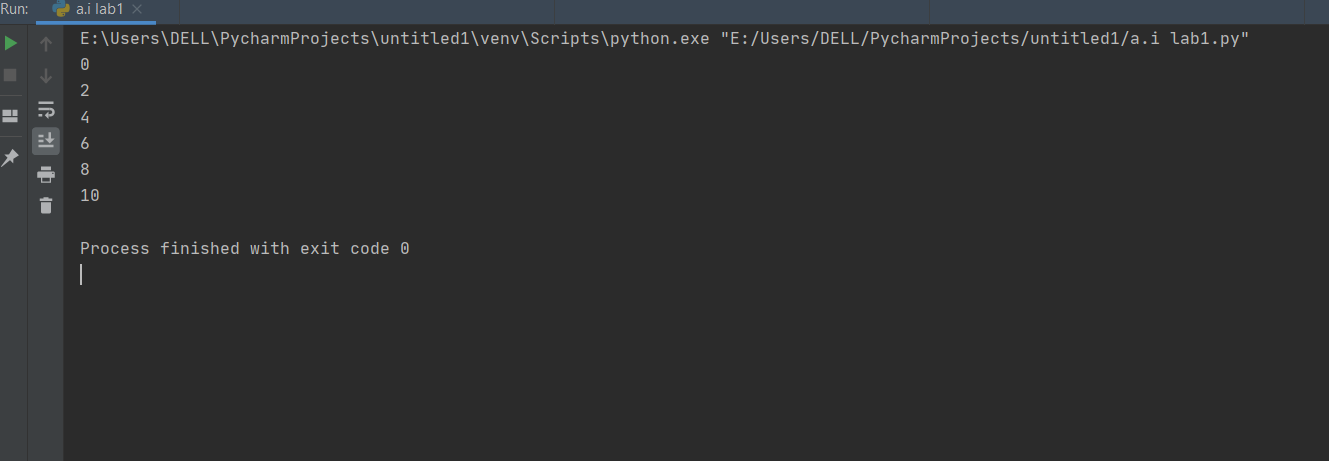
**Output:**

  
4. Create a loop that counts all even numbers to 10

**Source Code:**

for i in range(0,11,2):  
 print(i)

**Output**:

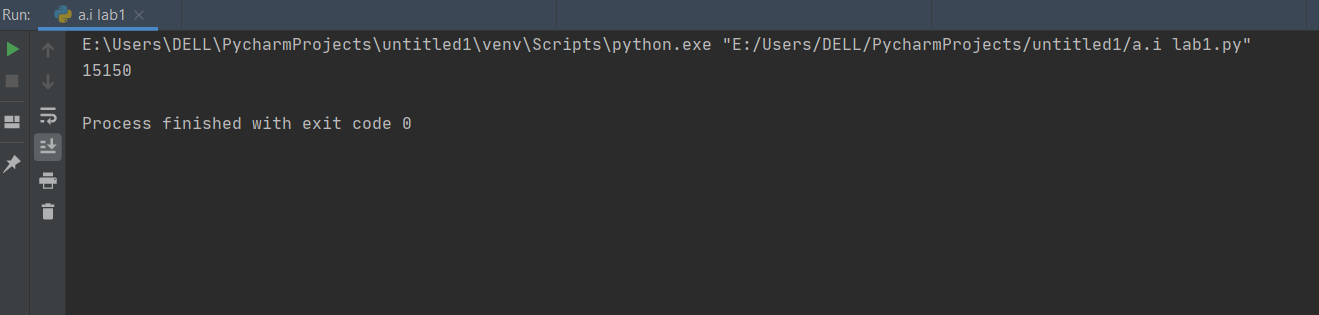


5. CrSeate a loop that sums the numbers from 100 to 200

Source Code:

sum = 0  
for i in range(100,201):  
 sum += i  
print(sum)

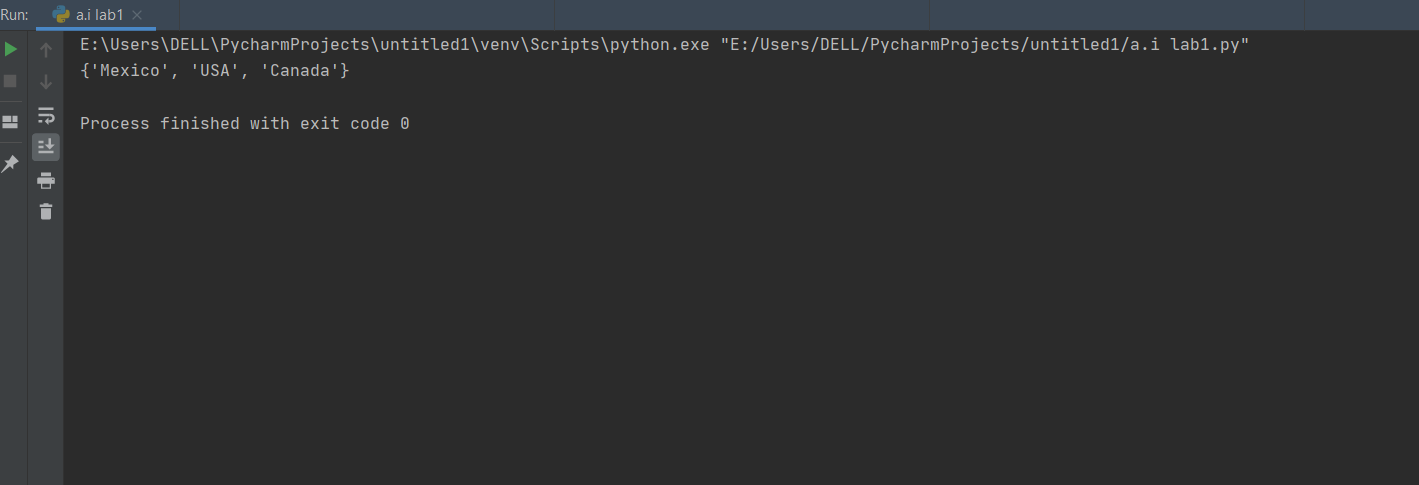
**Output:**



(iii) Try the exercise below:

1. Make a program that lists the countries in the set below using a while loop.

|  |
| --- |
| clist = ["Canada","USA","Mexico"] |
| **Source Code:**  clist=["Canada","USA","Mexico"] x=0 while x<len(clist):  print(set(clist))  x+=3  **Output:** |



2.What’s the difference between a while loop and a for loop?

**Answer:**

The 'for' loop used only when we already knew the number of iterations. The 'while' loop used only when the number of iteration are not exactly known. If the condition is not put up in 'for' loop, then loop iterates infinite times. If the condition is not put up in 'while' loop, it provides compilation error.

3.Can you sum numbers in a while loop?

**Answer:**

You can also use the while loop to calculate the sum and average of n numbers. ... Add the current value of n to sum variable. Also, decrement n by 1 in while loop body.

3. Can a for loop be used inside a while loop

**Answer:**

Any expression or variable can be a loop condition, not just comparisons: the condition is evaluated ... The for loop is more complex, but it's also the most commonly used loop. ... Such variables are visible only inside the loop.