COLLEGE OF ENGINEERING CHERTHALA

LAB RECORD

20MCA131 – PROGRAMMING LAB

CERTIFICATE

This is certified to be bonafide works of Mr./Ms. …………………………..………………….., In the class ………………...,

Reg. No. ……………………., of College of Engineering Cherthala, during the academic year 2022-23.

Teacher In Charge **External Examiner**

Internal Examiner

INDEX

Program No.1

Arithmetic operations

Aim :- To write a python program to perform to perform all the arithmetic operations

Algorithm :-

1: Ask the user to input two numbers.

2: Store the inputted numbers in separate variables.

3: Ask the user which arithmetic operation they want to perform (+, -, *, /, ** for exponentiation, // for floor division and % for modulo).

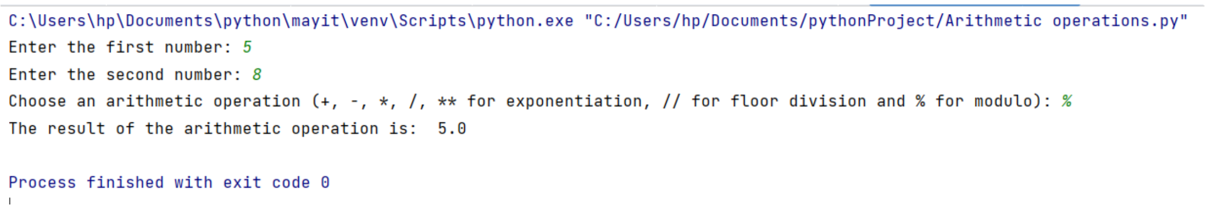
3: Based on the user's choice of operation, perform the corresponding arithmetic operation on the two inputted numbers.

4: Print the result of the arithmetic operation.

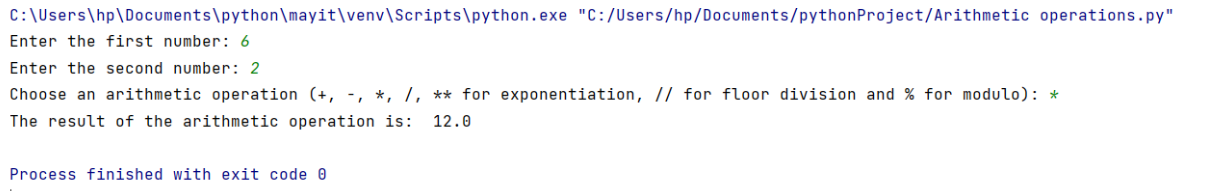
Program :-

```
# Ask the user to input two numbers

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

# Ask the user which arithmetic operation they want to perform

operation = input("Choose an arithmetic operation (+, -, *, /, ** for exponentiation, // for floor division and % for modulo): ")

# Based on the user's choice of operation, perform the corresponding arithmetic operation on the two inputted numbers

if operation == "+":

   result = num1 + num2

elif operation == "-":

   result = num1 - num2

elif operation == "*":

   result = num1 * num2

elif operation == "/":

   result = num1 / num2

elif operation == "**":
    result = num1 ** num2
elif operation == "//":
    result = num1 // num2
elif operation == "%":
    result = num1 % num2
else:
    print("Invalid input")
# Print the result of the arithmetic operation
print("The result of the arithmetic operation is: ", result)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Arithmetic operations.py"
Enter the first number: 5
Enter the second number: 8
Choose an arithmetic operation (+, -, *, /, ** for exponentiation, // for floor division and % for modulo): %
The result of the arithmetic operation is:  5.0

Process finished with exit code 0
```



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Arithmetic operations.py"
Enter the first number: 6
Enter the second number: 2
Choose an arithmetic operation (+, -, *, /, ** for exponentiation, // for floor division and % for modulo): *
The result of the arithmetic operation is:  12.0

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.2

Swap two numbers

Aim :- To write a python program to swap two numbers (with and without using another variable.)

Algorithm :-

1.Assign the value 10 to a.

2.Assign the value 5 to b.

3.Reassign the value 5 to a.

4.Reassign the value 10 to b.

5.Print the value of a before swapping.

6.Print the value of b before swapping.

7.Assign the value of a to temp.

8.Assign the value of b to a.

9.Assign the value of temp to b.

10.Print the value of a after swapping.

11.Print the value of b after swapping.

12.Assign the value 10 to a.

13.Assign the value 5 to b.

14.Print the value of a before swapping.

15.Print the value of b before swapping.

16.Add the values of a and b, and assign the result to a.

17.Subtract the original value of b from the new value of a, and assign the result to b.

18.Subtract the original value of b from the new value of a, and assign the result to a.

19.Print the value of a after swapping.

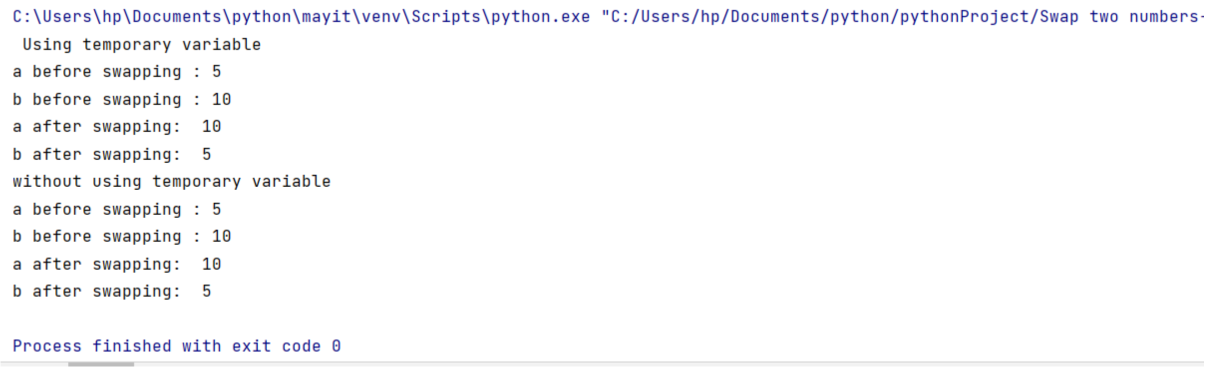
20.Print the value of b after swapping.

Program :-

```
print(" Using temporary variable")

a = 10

b = 5
```

```
a=5

b=10

print("a before swapping :",a)

print("b before swapping :",b)

temp = a

a = b

b = temp

print("a after swapping: ", a)

print("b after swapping: ", b)

print("without using temporary variable")

a=5

b=10

print("a before swapping :",a)

print("b before swapping :",b)

a = a + b

b = a - b

a = a - b

print("a after swapping: ", a)

print("b after swapping: ", b)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/python/pythonProject/Swap two numbers·
 Using temporary variable
a before swapping : 5
b before swapping : 10
a after swapping:  10
b after swapping:  5
without using temporary variable
a before swapping : 5
b before swapping : 10
a after swapping:  10
b after swapping:  5

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.3

Area of triangle

Aim :- To write a python program to find area of a triangle.

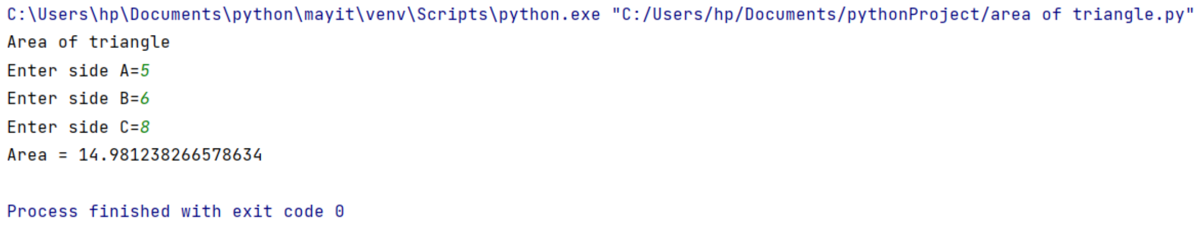
Algorithm :-

```
1:Read three sides of triangle ,say a,b,c
2:Calculate the semi perimeter using the formula s=(a+b+c)/2
3:Calculate the area using the formula (s*(s-a)*(s-b)*(s-c))**0.5
4:Print area
```

Program :-

```
print("Area of triangle")
a=int(input("Enter side A="))
b=int(input("Enter side B="))
c=int(input("Enter side C="))
s=(a+b+c)/2
area=(s*(s-a)*(s-b)*(s-c))**0.5
print("Area =",area)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/area of triangle.py"
Area of triangle
Enter side A=5
Enter side B=6
Enter side C=8
Area = 14.981238266578634

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.4

Largest number

Aim :- To write a python program to find the largest of three numbers

Algorithm :-

1.Take three input numbers a, b, and c from the user.

2.Check if a is greater than b and a is greater than c, then a is the largest number, print it.

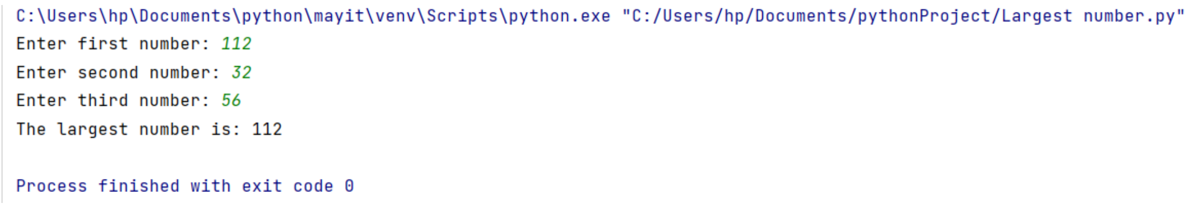
3.If the above condition is false, check if b is greater than a and b is greater than c, then b is the largest number, print it.

4.If both of the above conditions are false, then c is the largest number, print it.

Program :-

```
a = int(input("Enter first number: "))

b = int(input("Enter second number: "))

c = int(input("Enter third number: "))

if a > b and a > c:

   print("The largest number is:", a)

elif b > a and b > c:

   print("The largest number is:", b)

else:

   print("The largest number is:", c)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Largest number.py"
Enter first number: 112
Enter second number: 32
Enter third number: 56
The largest number is: 112

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.5

Read and print any n numbers

Aim :- To write a python program to read and print any n numbers

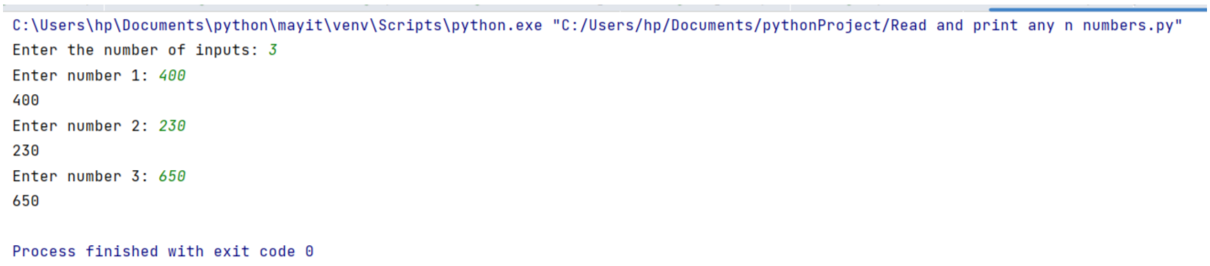
Algorithm :-

```
1.Take an input n from the user.
2.Use a for loop to iterate from 1 to n.
3.Within the loop, take an input number from the user and store it in a variable.
4.Print the number.
```

Program :-

```
n = int(input("Enter the number of inputs: "))
for i in range(1, n+1):
    num = int(input("Enter number {}: ".format(i)))
    print(num)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Read and print any n numbers.py"
Enter the number of inputs: 3
Enter number 1: 400
400
Enter number 2: 230
230
Enter number 3: 650
650

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.6

Multiplication table

Aim :- To write a python program to print multiplication table.

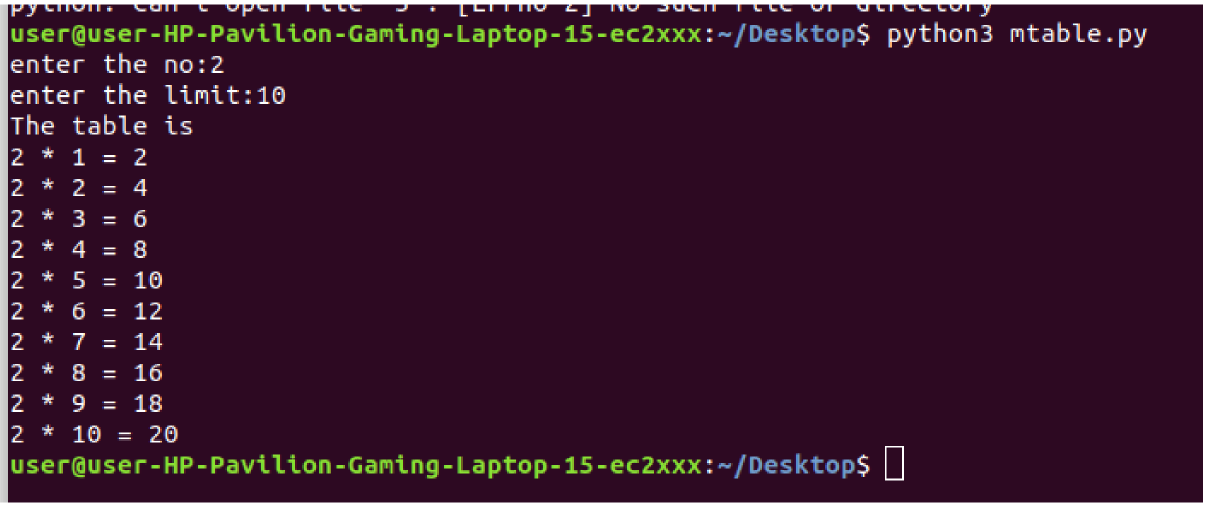
Algorithm :-

- Input the number for which the multiplication table is to be generated.
- Input the end value until which the table has to be generated.
- Repeat from i = 1 to end
- Display the table values in the given output format.(num * i = num*i)

Program :-

```
a=int(input("enter the no:"))
b=int(input("enter the limit:"))
print("The table is")
i=1
while(i<=b):
m=a*i
print(a,"*",i,"=",m)
i=i+1
```

Output :-



```
python: can't open file '5': [Errno 2] No such file or directory
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ python3 mtable.py
enter the no:2
enter the limit:10
The table is
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
2 * 10 = 20
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ 
```

Result :-

Program executed and output obtained successfully.

Program No.7

Prime number within a range

Aim :- To write a python program to print prime numbers within a range.

Algorithm :-

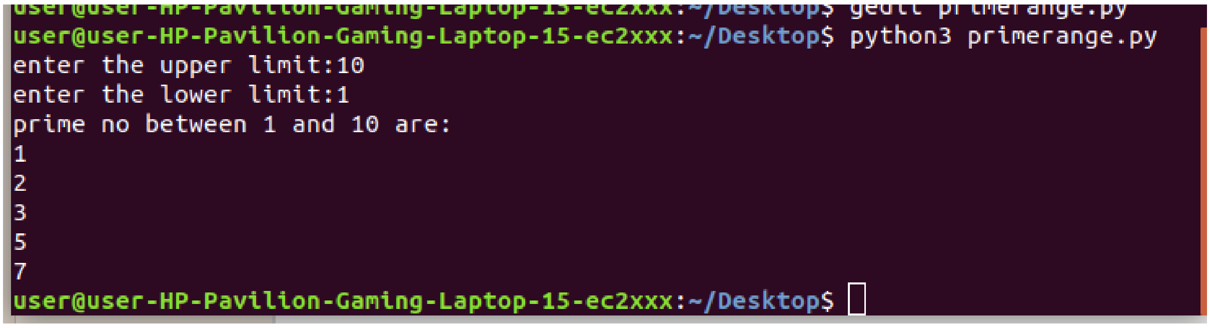
Step 1: Loop through all the elements in the given range.

- Step 2: Check for each number if it has any factor between 1 and itself.
- Step 3: If yes, then the number is not prime, and it will move to the next number.
- Step 4: If no, it is the prime number, and the program will print it and check for the next number.
- Step 5: The loop will break when it is reached to the upper value

Program :-

```
upper=int(input("enter the upper limit:"))
lower=int(input("enter the lower limit:"))
print("prime no between",lower,"and",upper,"are:")
for i in range(lower,(upper+1)):
f=0
j=2
while(j<=(i/2)):
if((i%j)==0):
f=1
break
j=j+1
if(f==0):
print(i)
```

Output :-



```
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ gedit primerange.py
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ python3 primerange.py
enter the upper limit:10
enter the lower limit:1
prime no between 1 and 10 are:
1
2
3
5
7
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ 
```

Result :-

Program executed and output obtained successfully.

Program No.8

First n prime numbers

Aim :- To write a python program.to print first n prime numbers.

Algorithm :-

1: Take input from the user for the number of prime numbers to be generated, n.

2: Initialize a variable, num, to 2, which is the first prime number.

3:Initialize a variable, count, to 0, which counts the number of prime numbers generated so far.

4:Repeat the following steps until count is equal to n:

a. Set a flag, is_prime, to True.

b. Check if num is divisible by any number from 2 to num-1.

i. If it is, set is_prime to False and break out of the loop.

ii. If not, continue checking for the next number.

c. If is_prime is True, print the number and increment count by 1.

d. Increment num by 1 and repeat from step 4a.

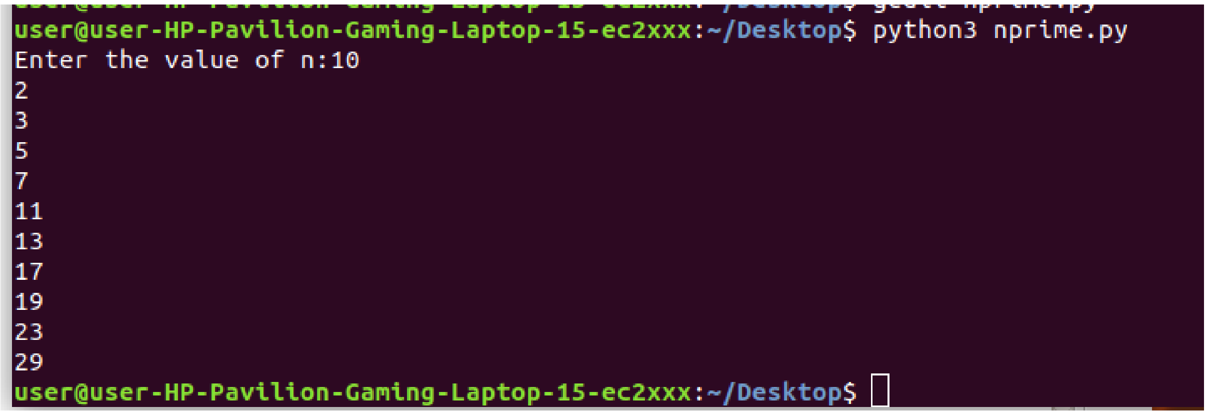
Program :-

```
num=int(input("Enter the value of n:"))

i=2

k=1

while(k<=num):

flag=0

j=2

while(j<=(i/2)):

if((i%j)==0):

flag=1

break

j=j+1

if(flag==0):

print(i)
```

k=k+1

i=i+1

Output :-



```
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ gedit nprime.py
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ python3 nprime.py
Enter the value of n:10
2
3
5
7
11
13
17
19
23
29
user@user-HP-Pavilion-Gaming-Laptop-15-ec2xxx:~/Desktop$ 
```

Result :-

Program executed and output obtained successfully.

Program No.9

Perfect or not

Aim :- To write a python program to check whether a number is perfect or not.

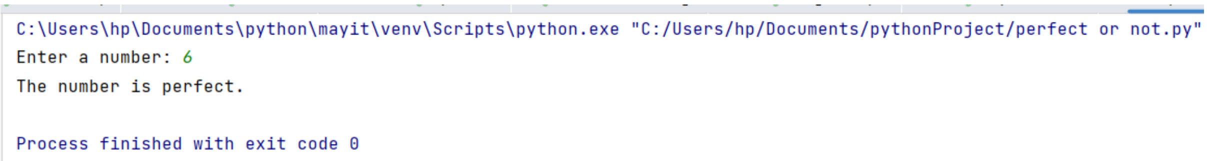
Algorithm :-

1. Take an input integer number from the user.
2. Initialize a variable called sum to 0.
3. Loop through all the numbers from 1 to the number (exclusive) using a for loop.
4. If the number is a factor of the input number, add it to the sum.
5. After the loop ends, check whether the sum is equal to the input number.If the sum is equal to the input number, print "The number is perfect." Otherwise, print "The number is not perfect

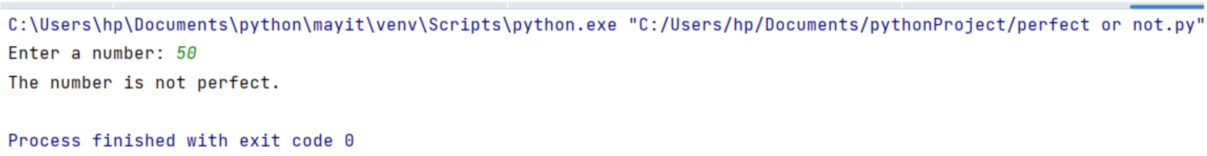
Program :-

```
num = int(input("Enter a number: "))
sum = 0

for i in range(1, num):
   if num % i == 0:
      sum += i

if sum == num:
   print("The number is perfect.")
else:
   print("The number is not perfect.")
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/perfect or not.py"
Enter a number: 6
The number is perfect.

Process finished with exit code 0
```



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/perfect or not.py"
Enter a number: 50
The number is not perfect.

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.10

Sum of n numbers using function

Aim :- To write a python program to print the sum of first n numbers using function

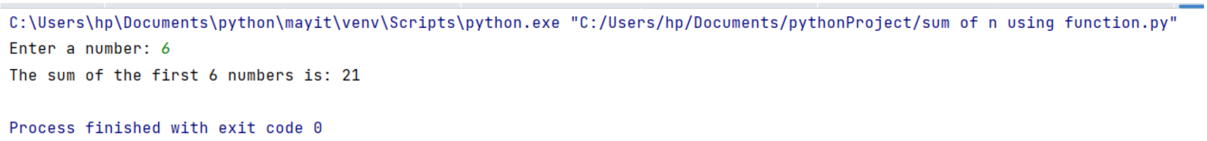
Algorithm :-

1. Define a function called sum_of_first_n that takes an integer n as an argument.
2. Initialize a variable called sum to 0.
3. Loop through all the numbers from 1 to n (inclusive) using a for loop.
4. Add each number to the sum.
5. After the loop ends, return the sum from the function.
6. Take an input integer n from the user.
7. Call the sum_of_first_n function with n as an argument and store the returned value in a variable called result.
8. Print the value of result to display the sum of the first n numbers.

Program :-

```
def sum_of_first_n(n):
    sum = 0
    for i in range(1, n+1):
        sum += i
    return sum

n = int(input("Enter a number: "))
result = sum_of_first_n(n)
print("The sum of the first", n, "numbers is:", result)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/sum of n using function.py"
Enter a number: 6
The sum of the first 6 numbers is: 21

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.11

Palindrome or not

Aim :- To write a python program to check whether a number is palindrome or not using function

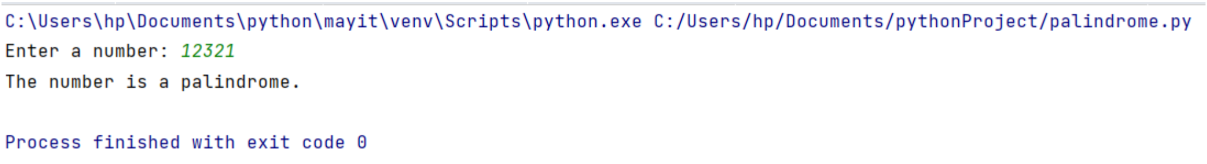
Algorithm :-

1. Define a function called is_palindrome that takes an integer n as an argument.
2. Convert the integer n to a string using the str() function and store it in a variable called n_str.
3. Reverse the string n_str using slicing and store it in a variable called n_rev.
4. Compare the original string n_str with the reversed string n_rev.
5. If the two strings are equal, return True from the function. Otherwise, return False.
6. Take an input integer n from the user.
7. Call the is_palindrome function with n as an argument and store the returned value in a variable called result.
8. If result is True, print "The number is a palindrome." Otherwise, print "The number is not a palindrome."

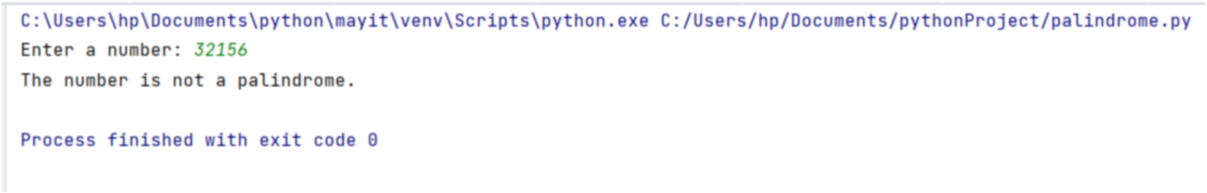
Program :-

```
def is_palindrome(n):
    n_str = str(n)
    n_rev = n_str[::-1]
    if n_str == n_rev:
        return True
    else:
        return False

n = int(input("Enter a number: "))
result = is_palindrome(n)
if result:
    print("The number is a palindrome.")

else:
    print("The number is not a palindrome.")
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe C:/Users/hp/Documents/pythonProject/palindrome.py
Enter a number: 12321
The number is a palindrome.

Process finished with exit code 0
```



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe C:/Users/hp/Documents/pythonProject/palindrome.py
Enter a number: 32156
The number is not a palindrome.

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.12

Binary to decimal

Aim :- To write a python program to convert binary number to decimal number

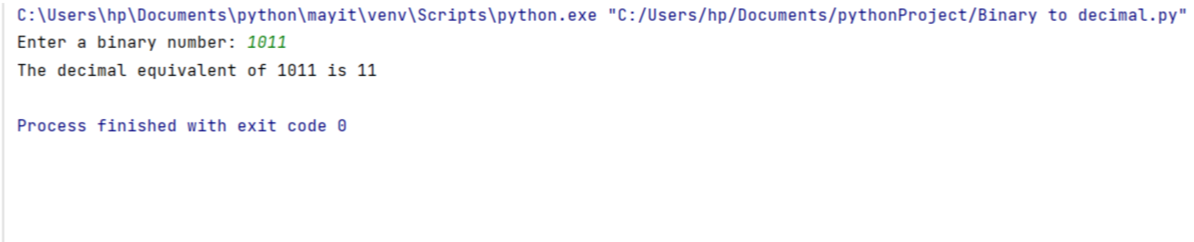
Algorithm :-

1. Take an input binary number from the user as a string.
2. Initialize a variable called decimal to 0.
3. Loop through each character in the binary string using a for loop.
4. For each character, multiply the current value of decimal by 2 and add the integer value of the character to it.
5. After the loop ends, the final value of decimal is the decimal equivalent of the binary number.
6. Print the value of decimal.

Program :-

```
binary = input("Enter a binary number: ")
decimal = 0

for digit in binary:
   decimal = decimal*2 + int(digit)

print("The decimal equivalent of", binary, "is", decimal)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Binary to decimal.py"
Enter a binary number: 1011
The decimal equivalent of 1011 is 11

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.13

Fibonacci using recursion

Aim :- To write a python program to print Fibonacci series of first n numbers using function(recursion)

Algorithm :-

1.Define a function called fibonacci that takes an integer n as an argument.

2.If n is 0 or 1, return n.

3.Otherwise, return the sum of the n-1th and n-2th Fibonacci numbers.

4.Take an input integer n from the user.

5.Loop through all the numbers from 0 to n-1 (inclusive) using a for loop.

6.For each number, call the fibonacci function with that number as an argument and print the returned value.

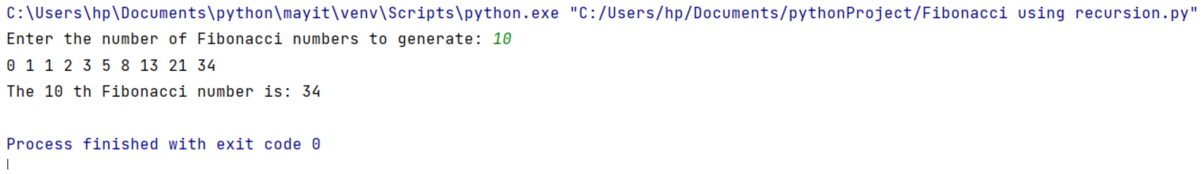
7.Call the fibonacci function with n as an argument and store the returned value in a variable called result.

8.Print the value of result to display the nth Fibonacci number

Program :-

```
def fibonacci(n):
   if n == 0 or n == 1:
      return n
   else:
      return fibonacci(n-1) + fibonacci(n-2)

n = int(input("Enter the number of Fibonacci numbers to generate: "))
for i in range(n):
   print(fibonacci(i), end=' ')

print("\nThe", n, "th Fibonacci number is:", fibonacci(n-1))
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Fibonacci using recursion.py"
Enter the number of Fibonacci numbers to generate: 10
0 1 1 2 3 5 8 13 21 34
The 10 th Fibonacci number is: 34

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.14

Factorial using recursion

Aim :- To write a python program to find the factorial of a number using recursion

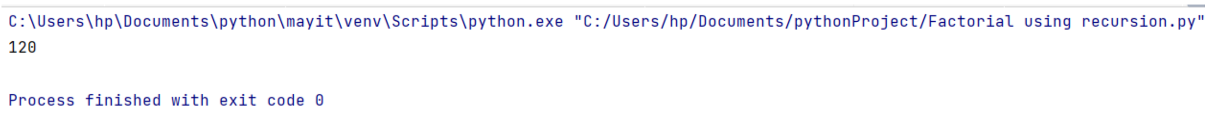
Algorithm :-

```
1:Define a function called "factorial" that takes a single parameter, "n", representing the number whose factorial is to be calculated.
2:Inside the "factorial" function, add a base case that checks whether "n" is equal to 0 or 1. If it is, return 1, as the factorial of 0 or 1 is always 1.
3:If the base case is not triggered, call the "factorial" function recursively with "n-1" as the parameter and store the result in a variable called "result".
4:Multiply "n" by "result" and return the result of this multiplication
```

Program :-

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        result = factorial(n-1)
        return n * result
print(factorial(5))
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Factorial using recursion.py"
120

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.15

Calculator using function

Aim :- To write a python program to create a calculator using function

Algorithm :-

1:Define a function called "add" that takes two numbers as inputs and returns their sum.

2:Define a function called "subtract" that takes two numbers as inputs and returns their difference.

3:Define a function called "multiply" that takes two numbers as inputs and returns their product.

4:Define a function called "divide" that takes two numbers as inputs and returns their quotient.

5:Define a function called "power" that takes two numbers as inputs and returns the first number raised to the power of the second number.

6:Define a function called "calculator" that takes three inputs: the first number, the operator, and the second number.

7:Inside the "calculator" function, use a conditional statement to determine which operation to perform based on the operator input.

8:Call the appropriate function based on the operator input and return the result.

9:Define a variable called "num1" and ask the user to input the first number.

10:Define a variable called "operator" and ask the user to input the operator.

11:Define a variable called "num2" and ask the user to input the second number.

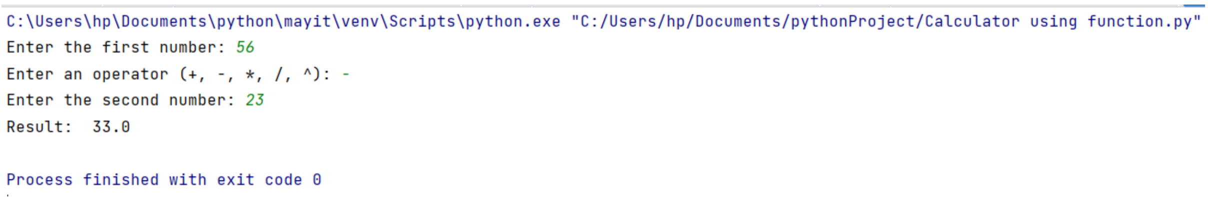
Call the "calculator" function with the three user inputs as arguments and print the result.

Program :-

```
def add(num1, num2):
   return num1 + num2
def subtract(num1, num2):
   return num1 - num2
def multiply(num1, num2):
   return num1 * num2
def divide(num1, num2):
   return num1 / num2
```

```
def power(num1, num2):
    return num1 ** num2
def calculator(num1, operator, num2):
    if operator == "+":
        return add(num1, num2)
    elif operator == "-":
        return subtract(num1, num2)
    elif operator == "*":
        return multiply(num1, num2)
    elif operator == "/":
        return divide(num1, num2)
    elif operator == "^":
        return power(num1, num2)
    else:
        return "Invalid operator"
num1 = float(input("Enter the first number: "))
operator = input("Enter an operator (+, -, *, /, ^): ")
num2 = float(input("Enter the second number: "))
result = calculator(num1, operator, num2)
print("Result: ", result)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Calculator using function.py"
Enter the first number: 56
Enter an operator (+, -, *, /, ^): -
Enter the second number: 23
Result:  33.0

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.16

Power using recursion

Aim :- To write a python program to find the power of number using recursion

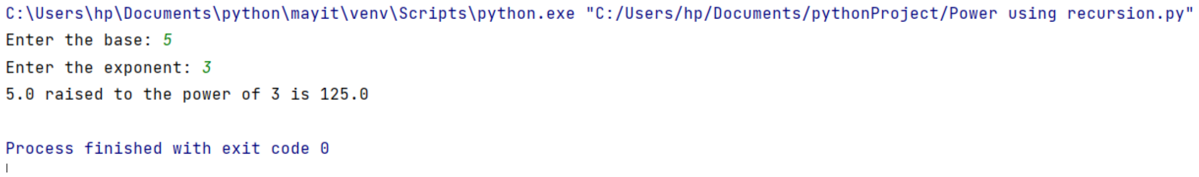
Algorithm :-

```
1:Define a function called "power" that takes two inputs: the base and the exponent.
2:Inside the "power" function, use a conditional statement to handle the base cases of the recursion: if the exponent is 0, return 1; if the exponent is 1, return the base.
3:If the base cases are not met, recursively call the "power" function with the base and exponent decremented by 1, and multiply the result by the base.
4:Return the result of the recursive call.
```

Program :-

```
def power(base, exponent):
    if exponent == 0:
        return 1
    elif exponent == 1:
        return base
    else:
        return base * power(base, exponent-1)
base = float(input("Enter the base: "))
exponent = int(input("Enter the exponent: "))
result = power(base, exponent)
print(base, "raised to the power of", exponent, "is", result)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Power using recursion.py"
Enter the base: 5
Enter the exponent: 3
5.0 raised to the power of 3 is 125.0

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.17

Count the occurrence of ‘a’

Aim :- To write a python program to read string and count the occurrence of ‘a’ in the string.

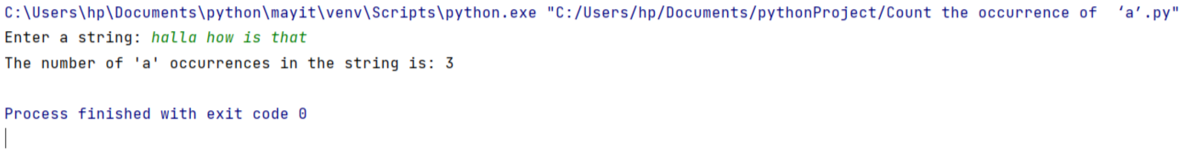
Algorithm :-

```
1:Take input from the user for the string to be checked.
2:Initialize a variable count to 0 to keep track of the number of occurrences of 'a'.
3:Loop through each character in the string using a for loop.
4:For each character, check if it is equal to 'a'.
5:If the character is 'a', increment the count variable by 1.
6:After the loop has finished, print the value of count to the user.
```

Program :-

```
string = input("Enter a string: ")

count = 0

for char in string:

    if char == 'a':

        count += 1

print("The number of 'a' occurrences in the string is:", count)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Count the occurrence of  ‘a’.py"
Enter a string: halla how is that
The number of 'a' occurrences in the string is: 3

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.18

Replace with $

Aim :- To write a python program to read string and replace all the occurrence of first character with ‘$’ except the first character

Algorithm :-

1. Prompt the user to enter a string

2. Read the input string and store it in a variable

3. Extract the first character of the string and store it in a variable

4. Extract the remaining characters of the string (excluding the first character) and replace all occurrences of the first character with a dollar sign

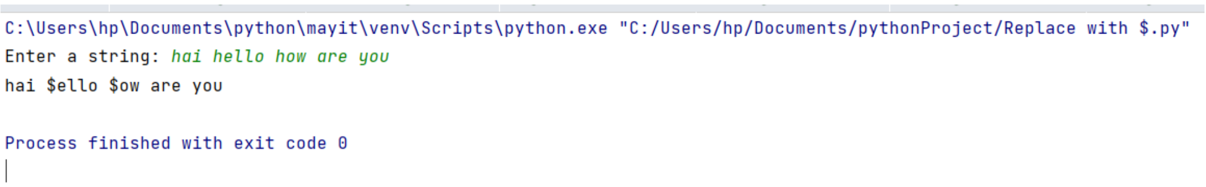
5. Concatenate the first character of the original string with the modified string

6. Print the modified string to the console

Program :-

```
string = input("Enter a string: ")

first_char = string[0]

new_string = first_char + string[1:].replace(first_char, '$')

print(new_string)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Replace with $.py"
Enter a string: hai hello how are you
hai $ello $ow are you

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.19

First & Last exchanged

Aim :- To write a python program to create a string from a given string where first and last characters are interchanged.

Algorithm :-

1.We initialize a variable start, which stores the first character of the string (string[0])

2.We initialize another variable end that stores the last character (string[-1])

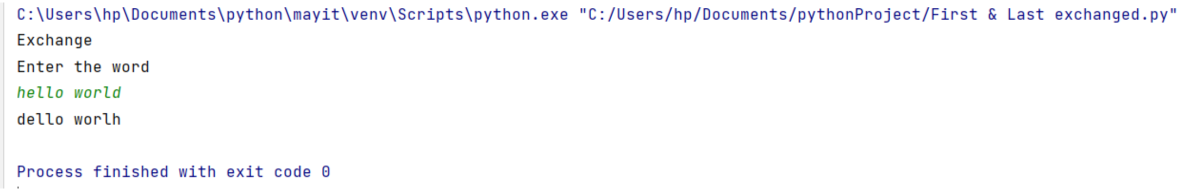
3.Then we will use string slicing, string[1:-1], this will access all the characters from the 2nd position excluding the last character.

4.Then we add these three as required forming a new string that has the first and last characters of the original string swapped. And then we will print it.

Program :-

```
print("Exchange")

def swap(a):
   s=a[0]
   e=a[-1]
   x=e+a[1:-1]+s
   print(x)

a = input("Enter the word\n")
swap(a)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/First & Last exchanged.py"
Exchange
Enter the word
hello world
dello worlh

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.20

Add ‘ing’ & ‘ly’

Aim :- To write a python program to add ‘ing’ at the end of a given string,if already ends with ing add ‘ly’.

Algorithm :-

1.Read the input string from the user.

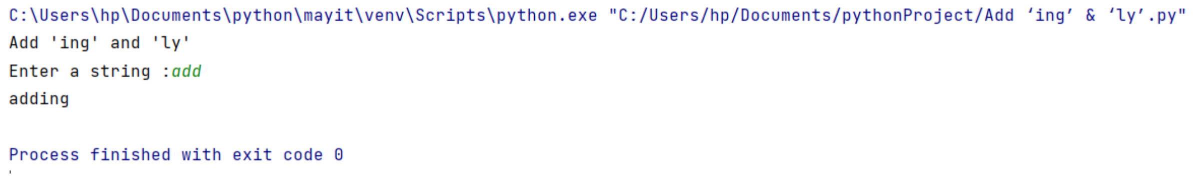
2.Check if the last three characters of the input string are 'ing'. If they are, append 'ly' to the input string and print the result.

3.If the last three characters of the input string are not 'ing', append 'ing' to the input string and print the result.

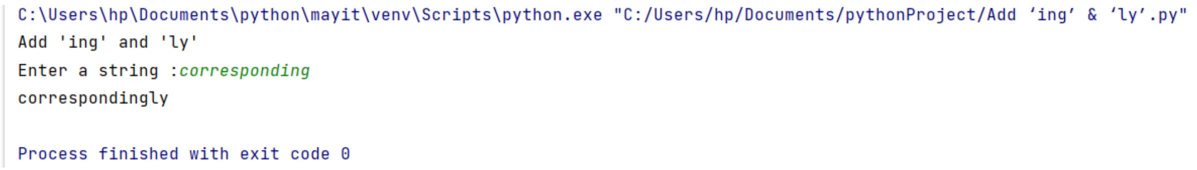
Program :-

```
print("Add 'ing' and 'ly'")

a=input("Enter a string :")

if(a[-1]=="g"):
   print(a+"ly")
else:
   print(a+"ing")
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Add ‘ing’ & ‘ly’.py"
Add 'ing' and 'ly'
Enter a string :add
adding

Process finished with exit code 0
```



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Add ‘ing’ & ‘ly’.py"
Add 'ing' and 'ly'
Enter a string :corresponding
correspondingly

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.21

Length of largest word

Aim :- To write a python program to accept a list of words and return the length of largest word

Algorithm :-

1. Take the number of elements in the list and store it in a variable.

2. Accept the values into the list using a for loop and insert them into the list.

3. First assume that the first element is the word with the longest length.

4. Then using a for loop and if statement, compare the lengths of the words in the list with the first element.

5. Store the name of the word with the longest length in a temporary variable.

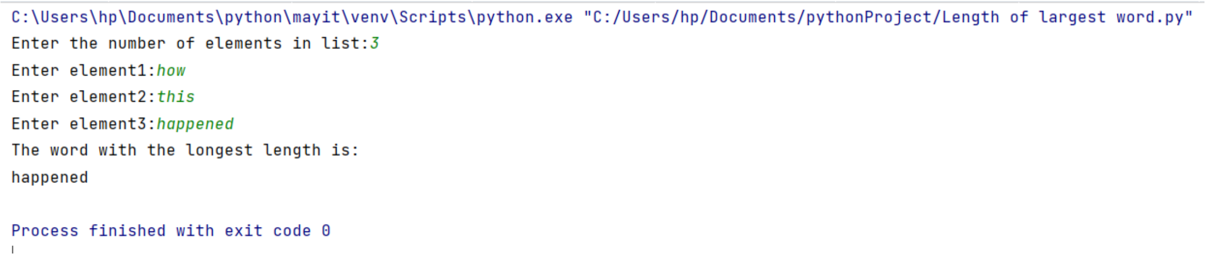
6. Display the word with the longest length

7. Exit.

Program :-

```
a=[]

n= int(input("Enter the number of elements in list:"))

for x in range(0,n):

   element=input("Enter element" + str(x+1) + ":")

   a.append(element)

max1=len(a[0])

temp=a[0]

for i in a:

   if(len(i)>max1):

      max1=len(i)

      temp=i

print("The word with the longest length is:")

print(temp)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Length of largest word.py"
Enter the number of elements in list:3
Enter element1:how
Enter element2:this
Enter element3:happened
The word with the longest length is:
happened

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.22

Count number of characters

Aim :- To write a python program to count the number of characters of a string

Algorithm :-

1.Define a string.

2.Define and initialize a variable count to 0.

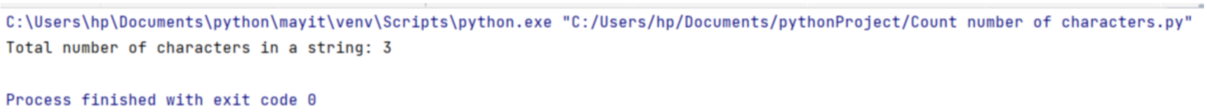
3.Iterate through the string till the end and for each character except spaces, increment the count by 1.

4.To avoid counting the spaces check the condition i.e. string[i] != ' '.

Program :-

```
string = "The";

count = 0;


for i in range(0, len(string)):

    if(string[i] != ' '):

        count = count + 1;

print("Total number of characters in a string: " + str(count));
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Count number of characters.py"
Total number of characters in a string: 3

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.23

String with space

Aim :- To write a python program to create a single string separated with space from two string

Algorithm :-

1:Prompt the user to enter the first string and store the input in a variable a.

2:Prompt the user to enter the second string and store the input in a variable b.

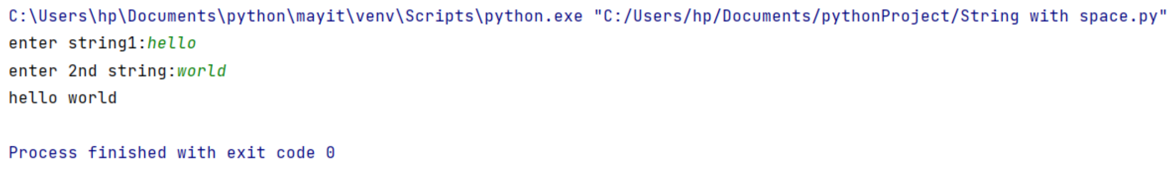
3:Concatenate the two strings a and b using the join() method and an empty string as the separator.

4:Print the concatenated string.

Program :-

```
a=input("enter string1:")

b=input("enter 2nd string:")

print("".join([a+""+b]))
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/String with space.py"
enter string1:hello
enter 2nd string:world
hello world

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.24

Alphabetic order name list

Aim :- To write a python program to create a name list of students in alphabetic order

Algorithm :-

1.input the number of names and declare a name.

2.input the names. Append the names to the name.

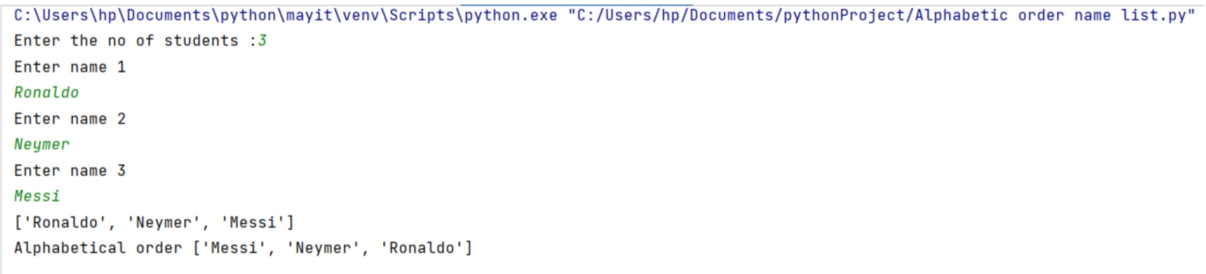
3.Use the default sort() method to sort the list.

4.print the names in the list by iterating over name using a for loop.

Program :-

```
name = []
n = int(input("Enter the no of students :"))

for i in range(1, n+1):
   print("Enter name", i, )
   item = str(input())
   name.append(item)

print(name)
name.sort()
print( "Alphabetical order",name)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Alphabetic order name list.py"
Enter the no of students :3
Enter name 1
Ronaldo
Enter name 2
Neymer
Enter name 3
Messi
['Ronaldo', 'Neymer', 'Messi']
Alphabetical order ['Messi', 'Neymer', 'Ronaldo']
```

Result :-

Program executed and output obtained successfully.

Program No.25

Student list using dictonary

Aim :- To write a python program to create a list of student record using dictionary.

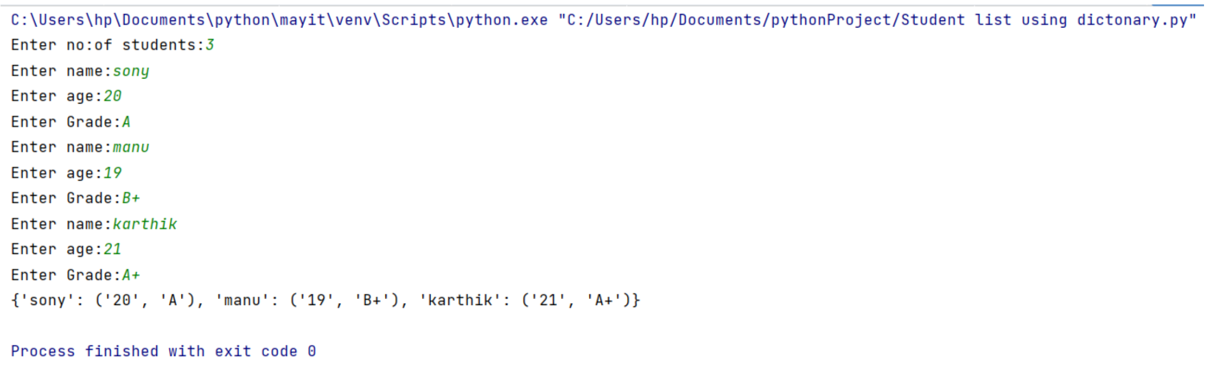
Algorithm :-

```
1:Initialize an empty dictionary called student.
2:Initialize an empty list called ls.
3:Prompt the user to enter the number of students n.
4:Loop through n times using a for loop.
5:Within the loop, prompt the user to enter the name of the student, age, and grade.
6:Add the name, age, and grade to the student dictionary as a tuple with the name as the key.
7:Print the student dictionary.
```

Program :-

```
student={}
ls=[]
n=int(input("Enter no:of students:"))


for i in range(0,n):
    name=input("Enter name:")
    age=input("Enter age:")
    grade=input("Enter Grade:")
    student[name]=age,grade


print(student)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Student list using dictonary.py"
Enter no:of students:3
Enter name:sony
Enter age:20
Enter Grade:A
Enter name:manu
Enter age:19
Enter Grade:B+
Enter name:karthik
Enter age:21
Enter Grade:A+
{'sony': ('20', 'A'), 'manu': ('19', 'B+'), 'karthik': ('21', 'A+')}

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.26

Ranklist in decending order

Aim :- To write a python program to print ranklist of students in descending order

Algorithm :-

1:Define a list of tuples, where each tuple represents a student's information. Each tuple should contain at least the student's name and their score.

2:Define a function that takes a student tuple as an argument and returns the student's score.

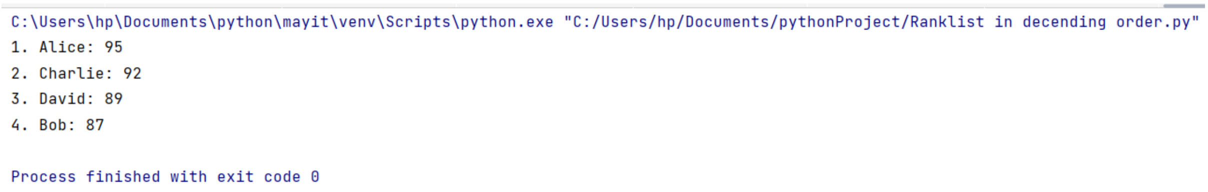
3:Sort the list of tuples in descending order of score using the sorted() function with the key argument set to the function defined in step 2.

4:Loop through the sorted list and print the rank, name, and score of each student using f-strings.

Program :-

```
students = [("Alice", 95), ("Bob", 87), ("Charlie", 92), ("David", 89)]

def get_score(student):

   return student[1]

sorted_students = sorted(students, key=get_score, reverse=True)

for i, student in enumerate(sorted_students):

   rank = i + 1

   name, score = student

   print(f"{rank}. {name}: {score}")
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Ranklist in decending order.py"
1. Alice: 95
2. Charlie: 92
3. David: 89
4. Bob: 87

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.27

Sort dictionary

Aim :- To write a python program to sort dictionary in ascending and descending order

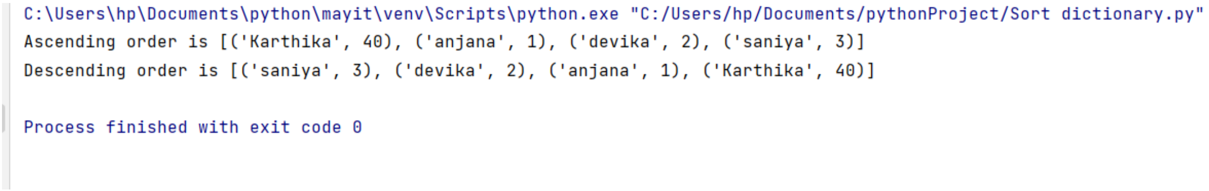
Algorithm :-

1. Define a function that takes a dictionary as input
2. Use the sorted() function with the items() method of the dictionary to sort the items by keys or values
3. If you want to sort by keys in ascending order, don't specify a key argument for sorted()
4. If you want to sort by keys in descending order, specify key=lambda x: -x[0] for sorted()
5. If you want to sort by values in ascending order, specify key=lambda x: x[1] for sorted()
6. If you want to sort by values in descending order, specify key=lambda x: -x[1] for sorted()
7. Assign the result to a new dictionary or a list of tuples

Program :-

```
y = {'Karthika': 40, 'devika':2, 'anjana': 1, 'saniya': 3 }

l = list (y.items())

l.sort()

print ('Ascending order is', l)

l= list (y.items())

l.sort (reverse =True)

print('Descending order is', l)

dict=dict(l)
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/pythonProject/Sort dictionary.py"
Ascending order is [('Karthika', 40), ('anjana', 1), ('devika', 2), ('saniya', 3)]
Descending order is [('saniya', 3), ('devika', 2), ('anjana', 1), ('Karthika', 40)]

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.28

Student list based on attendance

Aim :- To write a python program to print student list based on attendance

Algorithm :-

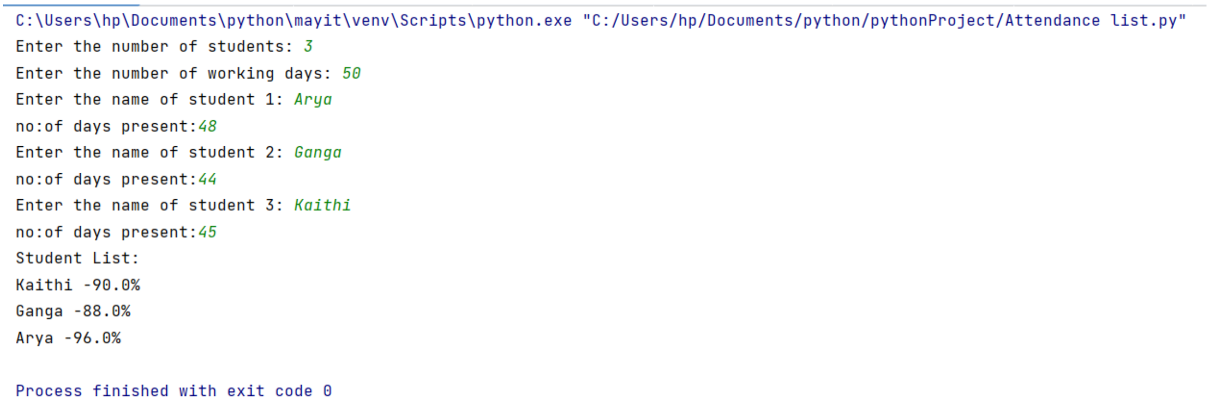
```
1.Ask the user to input the number of students and the number of present days.
2.Create an empty list called "student_list".
3.Use a for loop to iterate through the range of the number of students entered by the user.
4.Inside the loop, ask the user to enter the name of each student and the number of days they were present.
5.Calculate the attendance percentage for each student by dividing their number of present days by the total number of present days, then multiplying by 100.
6.Append a tuple containing the student's name and attendance percentage to the "student_list".
7.Sort the "student_list" in descending order by attendance percentage using the "sort()" method.
8.Use another for loop to iterate through the sorted "student_list".
9.Inside the loop, print each student's name and attendance percentage in the format "name - percentage%".
```

Program :-

```
num_students = int(input("Enter the number of students: "))
days = int(input("Enter the number of working days: "))


student_list = []


for i in range(num_students):
    name = input(f"Enter the name of student {i+1}: ")
    attendance = int(input("no:of days present:"))
    attendance_percentage = ((attendance / days) * 100)
    student_list.append((name, attendance_percentage))
    student_list.sort(reverse=True)
print("Student List:")
```

```
for student in student_list:

    print( f"{student[0]} -{student[1]}%")
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/python/pythonProject/Attendance list.py"
Enter the number of students: 3
Enter the number of working days: 50
Enter the name of student 1: Arya
no:of days present:48
Enter the name of student 2: Ganga
no:of days present:44
Enter the name of student 3: Kaithi
no:of days present:45
Student List:
Kaithi -90.0%
Ganga -88.0%
Arya -96.0%

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.29

Lambda function

Aim :- To write a python program to execute the following equation using lambda function a2-b2.

Algorithm :-

1. Define a lambda function x that takes two arguments 'a' and 'b' and returns the value of f(a, b) = a^2 - b^2.

2. Take user input for 'a' and 'b'.

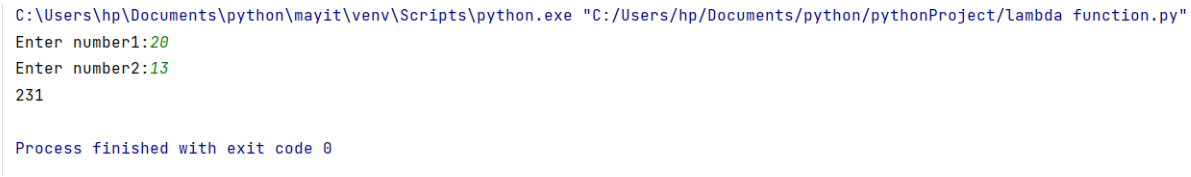
3. Compute the output of the lambda function x for the given inputs 'a' and 'b'.

4. Print the output of the lambda function.

Program :-

```
x=lambda a,b:a*a-b*b

a=int(input("Enter number1:"))

b=int(input("Enter number2:"))

print(x(a,b))
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/python/pythonProject/lambda function.py"
Enter number1:20
Enter number2:13
231

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.30

Find area and perimeter using lambda functions

Aim :- To write a python program to find area and perimeter of square and rectangle using lambda function.

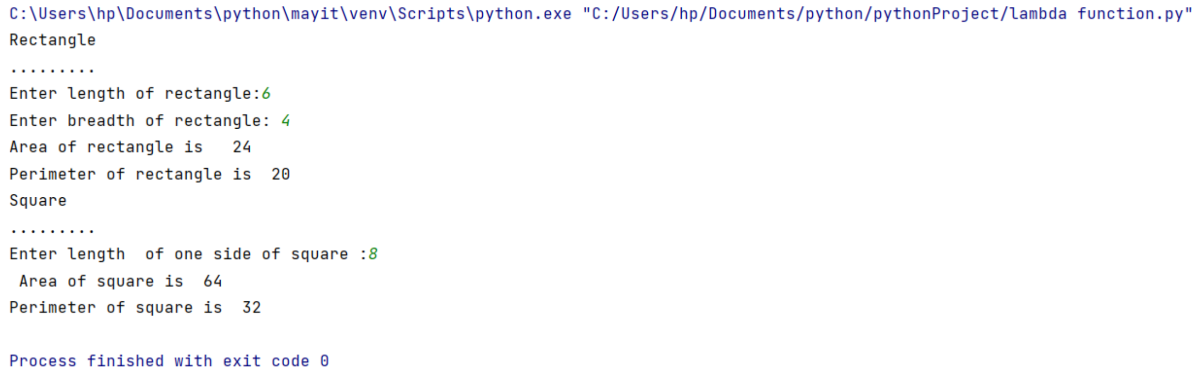
Algorithm :-

```
1.Print "Rectangle" and "........." to represent that the following calculations are for a rectangle.
2.Take input of length (l) and breadth (b) of the rectangle using the int() method.
3.Define a lambda function area that takes l and b as arguments and returns their product l*b.
4.Print the area of the rectangle using the lambda function defined above.
5.Define another lambda function per that takes l and b as arguments and returns the perimeter of the rectangle using the formula 2*(l+b).
6.Print the perimeter of the rectangle using the lambda function defined above.
7.Print "Square" and "........." to represent that the following calculations are for a square.
8.Take input of length (a) of one side of the square using the int() method.
9.Define a lambda function area that takes a as an argument and returns its square a*a.
10.Print the area of the square using the lambda function defined above.
11.Define another lambda function per that takes a as an argument and returns the perimeter of the square using the formula 4*a.
12.Print the perimeter of the square using the lambda function defined above.
```

Program :-

```
print("Rectangle")
print(".........")
l=int(input("Enter length of rectangle "))
b=int(input("Enter breadth of rectangle "))
area=lambda l1,b1:l1*b1
print("Area of rectangle is  ",area(l,b))


per=lambda l,b:2*(l+b)
print("Perimeter of rectangle is ",per(l,b))
print("Square")

print(".........")

a=int(input("Enter length  of one side of square "))

area=lambda a:a*a

print(" Area of square is ",area(a))

per=lambda a:4*a

print("Perimeter of square is ",per(a))
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/python/pythonProject/lambda function.py"
Rectangle
.........
Enter length of rectangle:6
Enter breadth of rectangle: 4
Area of rectangle is   24
Perimeter of rectangle is  20
Square
.........
Enter length  of one side of square :8
 Area of square is  64
Perimeter of square is  32

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.31

Rectangle class

Aim :- To write a python program to create a class Rectangle with attributes length and breadth and methods to find area and perimeter.

Algorithm :-

1.Define a class rect with the _init_ method that takes two arguments, breadth and length, and initializes the instance variables with these values.

2.Define a method area in the rect class that returns the product of the breadth and length instance variables.

3.Define a method per in the rect class that returns twice the sum of the breadth and length instance variables.

4.Prompt the user to enter the length of the rectangle and store it in the variable a.

5.Prompt the user to enter the breadth of the rectangle and store it in the variable b.

6.Create an object obj of the rect class with the a and b values.

7.Call the area method of the obj object and print the result with a message "Area of rectangle:".

8.Call the per method of the obj object and print the result with a message "Perimeter of rectangle:".

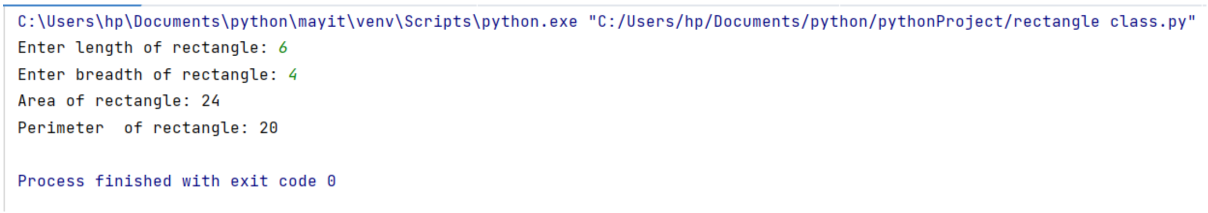
9.Exit.

Program :-

```
class rect:
    def __init__(self,breadth,length):
        self.breadth=breadth
        self.length=length


    def area(self):
        return self.breadth*self.length


    def per(self):
        return 2*(self.length+self.breadth)
```

```
a=int(input("Enter length of rectangle: "))
b=int(input("Enter breadth of rectangle: "))
obj=rect(a,b) # Creating an object 'obj' of class rect


print( "Area of rectangle:",obj.area())
print( "Perimeter  of rectangle:",obj.per())
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/python/pythonProject/rectangle class.py"
Enter length of rectangle: 6
Enter breadth of rectangle: 4
Area of rectangle: 24
Perimeter  of rectangle: 20

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.

Program No.32

Bank account class

Aim :- To write a python program to create Bank account class.

Algorithm :-

1.Define a class named Bank_Account.

2.In the _init_ method, initialize the account balance to 0 and print a welcome message.

3.Define a deposit method that prompts the user to enter an amount to be deposited, adds the amount to the account balance, and prints a message indicating the amount deposited.

4.Define a withdraw method that prompts the user to enter an amount to be withdrawn, checks if the account balance is greater than or equal to the requested amount, subtracts the requested amount from the account balance if it is, and prints a message indicating the amount withdrawn or an error message if the account balance is insufficient.

5.Define a display method that prints the current account balance.

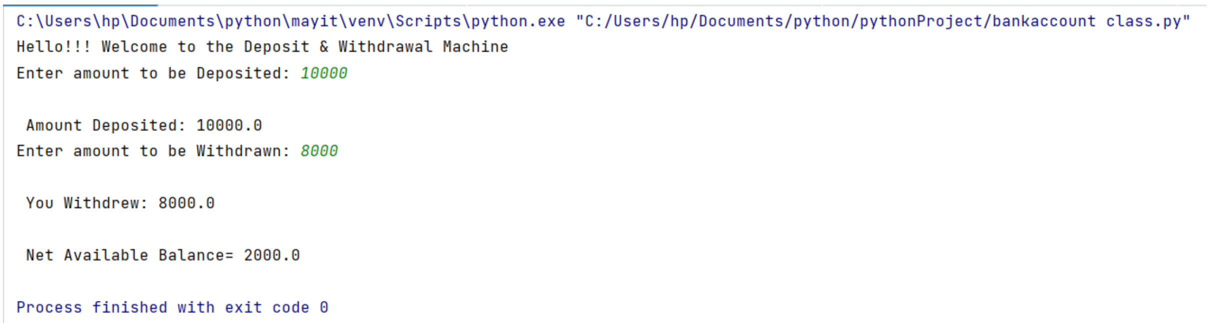
6.Create an instance of the Bank_Account class.

7.Call the deposit, withdraw, and display methods on the instance to interact with the bank account

Program :-

```
class Bank_Account:

    def __init__(self):

        self.balance=0

        print("Hello!!! Welcome to the Deposit & Withdrawal Machine")

    def deposit(self):

        amount=float(input("Enter amount to be Deposited: "))

        self.balance += amount

        print("\n Amount Deposited:",amount)


    def withdraw(self):

        amount = float(input("Enter amount to be Withdrawn: "))

        if( self.balance>=amount):

            self.balance=self.balance-amount

            print("\n You Withdrew:", amount)
            else:

                print("\n Insufficient balance ")


    def display(self):

        print("\n Net Available Balance=",self.balance)

# creating an object of class

s = Bank_Account()


# Calling functions with that class object

s.deposit()

s.withdraw()

s.display()
```

Output :-



```
C:\Users\hp\Documents\python\mayit\venv\Scripts\python.exe "C:/Users/hp/Documents/python/pythonProject/bankaccount class.py"
Hello!!! Welcome to the Deposit & Withdrawal Machine
Enter amount to be Deposited: 10000

 Amount Deposited: 10000.0
Enter amount to be Withdrawn: 8000

 You Withdrew: 8000.0

 Net Available Balance= 2000.0

Process finished with exit code 0
```

Result :-

Program executed and output obtained successfully.