

LABCYCLE 2

EXPERIMENT NO:1

FACTORIAL

Date: 12/12/2022

AIM: Program to find the factorial of a number.

ALGORITHM:

Step 1: Start.

Step 2: Input an integer number from the user.

Step 3: Initialize fact=1.

Step 4: Use for loop to multiply “fact” with all the numbers less than and equal to the number given by the user.

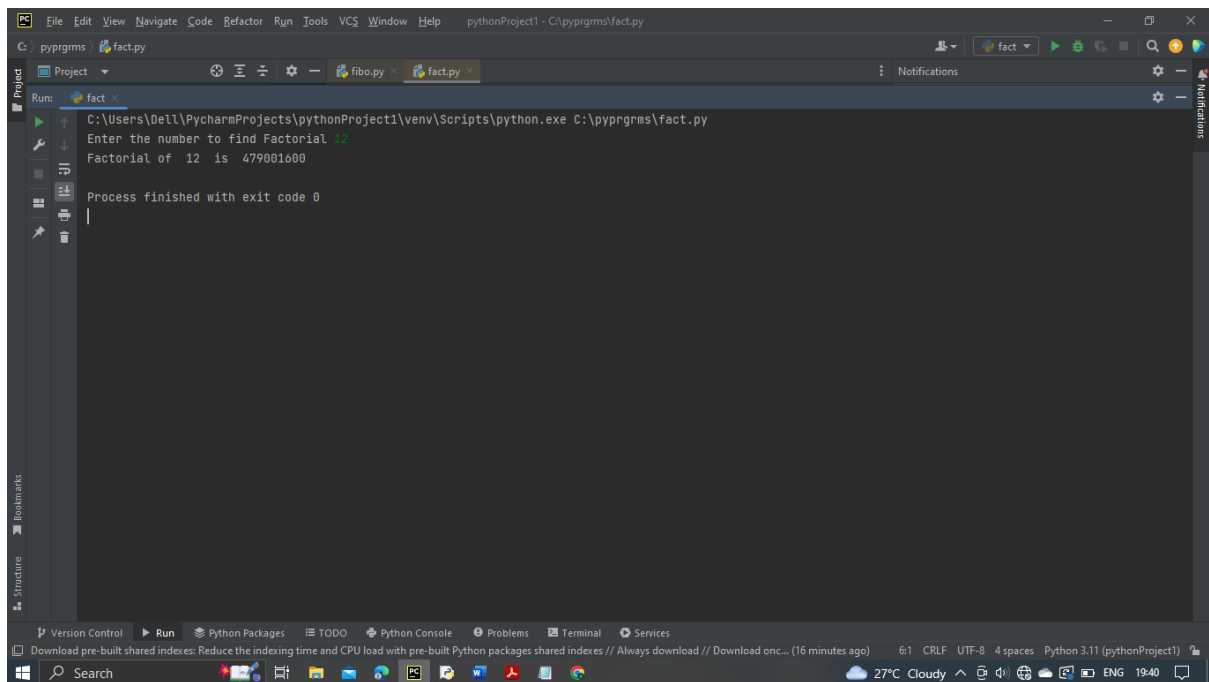
Step 5: Now, print the factorial of that number.

Step 6: Stop.

PROGRAM CODE:

```
def fact(num):  
    f=1  
    if num==0:  
        print("Factorial is : ",f)  
    elif num<0:  
        print("Can't find the factorial ")  
    else:  
        for i in range(1,num+1):  
            f=f*i  
        print("Factorial of ", num," is ",f)  
  
num=int(input("Enter the number to find Factorial "))  
fact(num)
```

OUTPUT:



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 is named 'pythonProject1' and the file being edited is 'fact.py'. The Run toolbar at the top shows a green play button and a status bar indicating 'fact' is running. The Run console on the left shows the command 'C:\Users\Dell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\fact.py'. The main editor area displays the output of the program: 'Enter the number to find Factorial', 'Factorial of 12 is 479001600', and 'Process finished with exit code 0'. The bottom status bar shows the Python version as 3.11 (pythonProject1) and the current time as 19:40.

```
C:\Users\Dell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\fact.py
Enter the number to find Factorial
Factorial of 12 is 479001600
Process finished with exit code 0
```

RESULT:

Program to find factorial has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:2

FIBONACCI SERIES

Date: 12/12/2022

AIM: Generate Fibonacci series of N terms.

ALGORITHM:

Step 1: Take the first two numbers of the series and the number of terms to be printed from the user.

Step 2: Print the first two numbers.

Step 3: Use a for loop to find the sum of the first two numbers and then proceed the fibonacci series.

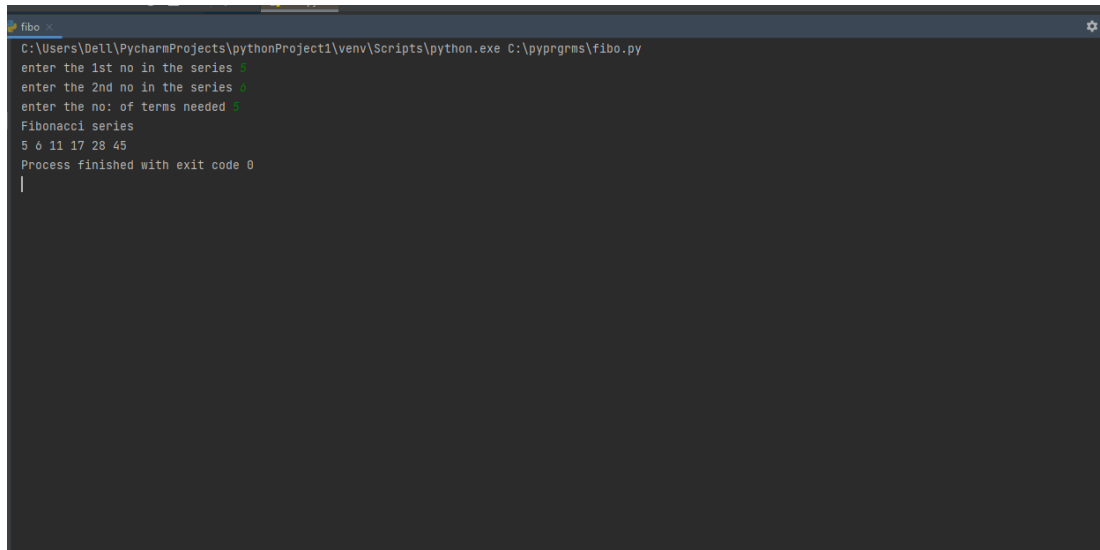
Step 4: Print the fibonacci series till $n-+1$.

Step 5: Stop.

PROGRAM CODE:

```
a=int(input("enter the 1st no in the series "))
b=int(input("enter the 2nd no in the series "))
n=int(input("enter the no: of terms needed "))
print("Fibonacci series ")
print(a,b, end=" ")
for i in range(2,n+1):
    f=a+b
    a=b
    b=f
    print(f, end=" ")
```

OUTPUT:

A screenshot of a terminal window titled 'fibonacci'. The window shows the execution of a Python script. The prompt is 'C:\Users\Deell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\fibonacci.py'. The user enters '5', '6', and '17' in response to prompts 'enter the 1st no in the series', 'enter the 2nd no in the series', and 'enter the no: of terms needed'. The output is 'Fibonacci series' followed by '5 6 11 17 28 45'. The terminal ends with 'Process finished with exit code 0' and a cursor on a new line.

```
C:\Users\Deell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\fibonacci.py
enter the 1st no in the series
enter the 2nd no in the series
enter the no: of terms needed
Fibonacci series
5 6 11 17 28 45
Process finished with exit code 0
```

RESULT:

Program to display fibonacci has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:3

SUM OF ITEMS IN A LIST

Date: 12/12/2022

AIM: Find the sum of all items in a list.

ALGORITHM:

Step 1: Start.

Step 2: Take the number of elements in the list and store it in a variable.

Step 3: Accept the values into the list using for loop and insert into the list.

Step 4: Initialize a variable s=0 to store the sum.

Step 5: Access each element in list using for loop and take sum.

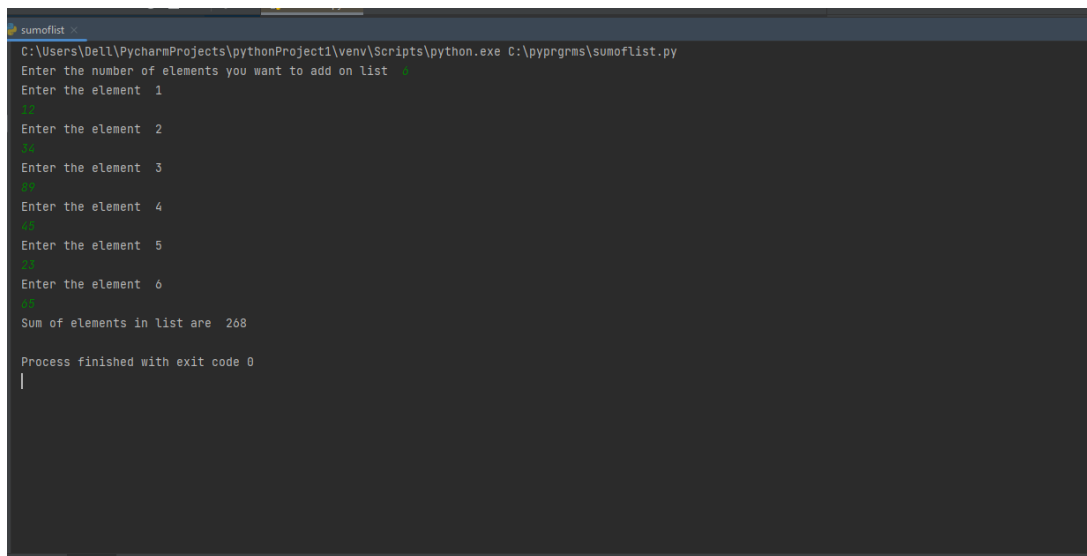
Step 6: Print s.

Step 7: Stop.

PROGRAM CODE:

```
list1=[]
len1=int(input("Enter the number of elements you want to add on list "))
for i in range(0,len1):
    print("Enter the element ",i+1)
    inp=int(input())
    list1.append(inp)
s=0
for i in list1:
    s=s+i
print("Sum of elements in list are ",s)
```

OUTPUT:

A screenshot of a terminal window titled 'sumoflist'. The terminal shows the execution of a Python script. The prompt is 'C:\Users\Deil\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\sumoflist.py'. The script prompts the user to 'Enter the number of elements you want to add on list', and the user enters '6'. The script then prompts the user to 'Enter the element' six times, with the user entering '1', '2', '3', '4', '5', and '6' respectively. The script then outputs 'Sum of elements in list are 268'. Finally, the terminal shows 'Process finished with exit code 0' and a cursor on a new line.

```
sumoflist
C:\Users\Deil\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\sumoflist.py
Enter the number of elements you want to add on list 6
Enter the element 1
2
Enter the element 2
3
Enter the element 3
4
Enter the element 4
5
Enter the element 5
6
Enter the element 6
Sum of elements in list are 268

Process finished with exit code 0
|
```

RESULT:

Program to find sum of elements in a list has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:4

EVEN FOUR DIGIT IS A PERFECT SQUARE

Date: 14/12/2022

AIM: Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

ALGORITHM:

Step 1: Start.

Step 2: Set the lower and upper range for four-digit number.

Step 3: Set the lower and upper range for the square number which result four-digit.

Step 4: Check for each digit is even or not.

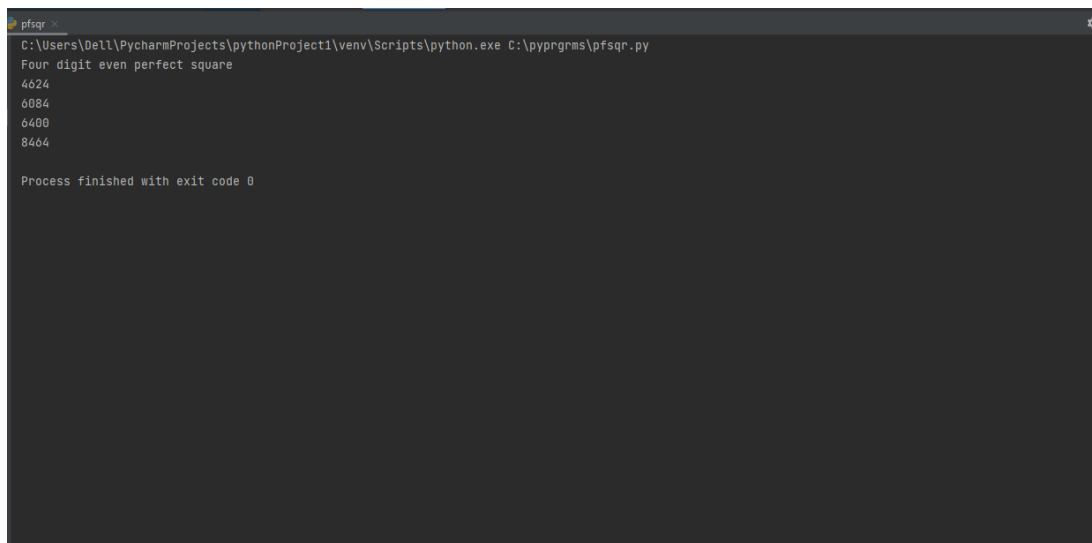
Step 5: Print the number.

Step 6: Stop.

PROGRAM CODE:

```
print("Four digit even perfect square ")
for i in range(1000,10000,1):
    for j in range(32,100,1):
        if i==j*j:
            string=str(i)
            if int(string[0])%2==0 and int(string[1])%2==0 and
                int(string[2])%2==0 and int(string[3])%2==0:
                print(i)
```

OUTPUT:



```
pfsqr
C:\Users\Dell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\pfsqr.py
Four digit even perfect square
4624
6084
6400
8464

Process finished with exit code 0
```

RESULT:

Program to find even four digit perfect square has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:5

NUMBER PYRAMID

Date: 19/12/2022

AIM: Display the given pyramid with step number accepted from user.

```
1
2 4
3 6 9
4 8 12 16
```

ALGORITHM:

Step 1: Start.

Step 2: Set the limit.

Step 3: Set the range of values.

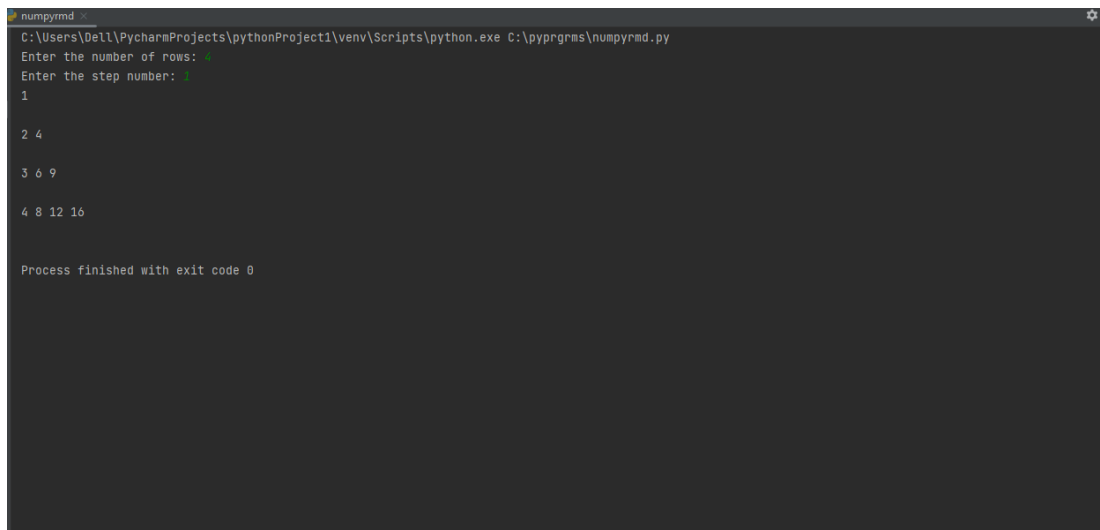
Step 4: Print product.

Step 5: Stop.

PROGRAM CODE:

```
rows=int(input("Enter the number of rows: "))
step=int(input("Enter the step number: "))
for i in range(1,rows+1,step):
    for j in range(1,i+1,step):
        print(i*j,end=" ")
    print("\n")
```

OUTPUT:



```
numpyrmd
C:\Users\De11\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\numpyrmd.py
Enter the number of rows: 4
Enter the step number: 1
1
2 4
3 6 9
4 8 12 16

Process finished with exit code 0
```

RESULT:

Program to display number pyramid has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:6

CHARACTER FREQUENCY

Date: 14/12/2022

AIM: Count the number of characters (character frequency) in a string.

ALGORITHM:

Step 1: Start.

Step 2: Define a string.

Step 3: Define an array freq with same size of the string.

Step 4: Two loops will be used to count the frequency of each character. Outer loop will be used to select a character and initialize element at corresponding index in array freq with 1.

Step 5: Inner loop will compare the selected character with rest of the character present in the string.

Step 6: If a match found ,increment element in freq by 1 and set the duplicated of selected character by '0' to mark them as visited.

Step 7: Finally, display the character and their corresponding frequencies by iterating through the array freq.

Step 8: Stop.

PROGRAM CODE:

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

```
freq=[None]*len(string)
```

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

```
    freq[i]=1
```

```
    for j in range(i+1,len(string)):
```

```
        if(string[i]==string[j]):
```

```
freq[i]=freq[i]+1

string=string[:j]+'0'+string[j+1:]

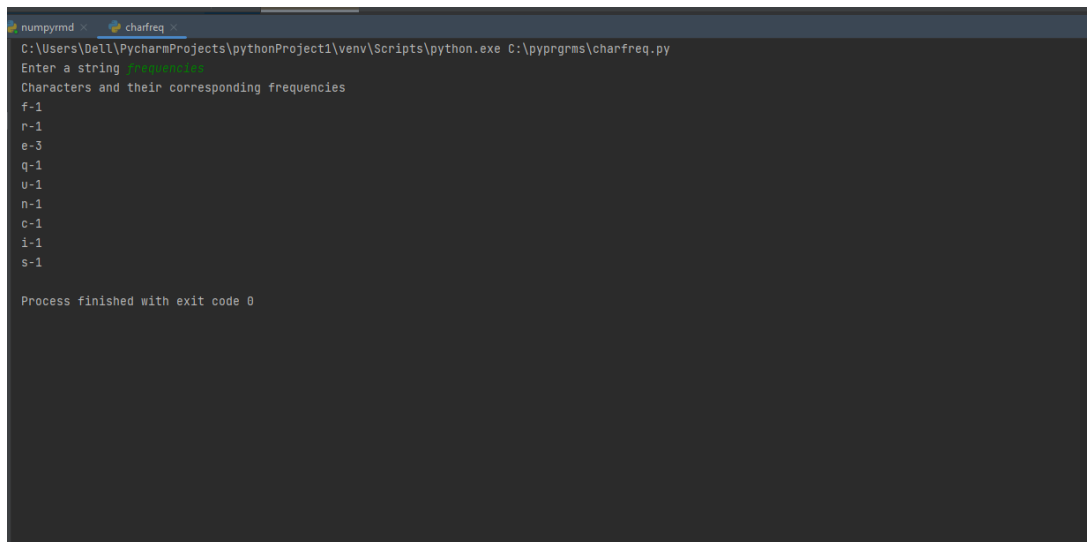
print("Characters and their corresponding frequencies")

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

    if(string[i]!=" " and string[i]!='0'):

        print(string[i]+"-"+str(freq[i]))
```

OUTPUT:



```
numpyrmd x charfreq x
C:\Users\De11\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\charfreq.py
Enter a string frequencies
Characters and their corresponding frequencies
f-1
r-1
e-3
q-1
u-1
n-1
c-1
i-1
s-1

Process finished with exit code 0
```

RESULT:

Program to print character frequency has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:7

ADDING “ly” AND “ing” TO A STRING

Date: 14/12/2022

AIM: Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’.

ALGORITHM:

Step 1: Start.

Step 2: Input a string and find the length of string.

Step 3: Check if length greater than 0:

If ing present in string , str+="ly".

Else:

str+="ing"

Step 4: Print string.

Step 5: Stop.

PROGRAM CODE:

```
def addstr(str):
```

```
    l=len(str)
```

```
    if l>0:
```

```
        if str[-3:]=="ing":
```

```
            str+="ly"
```

```
        else:
```

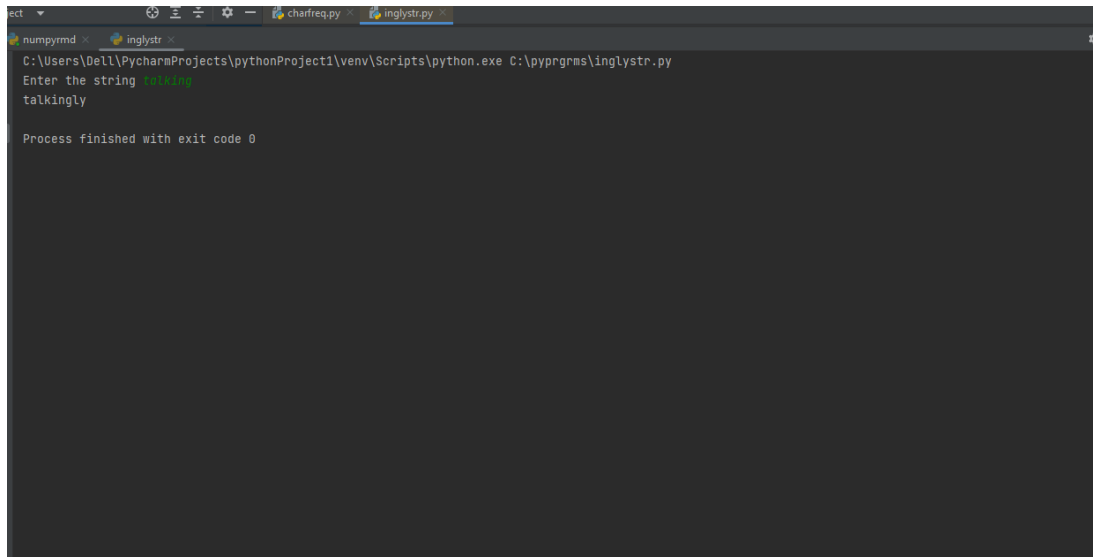
```
            str+="ing"
```

```
    print(str)
```

```
str=input("Enter the string ")
```

addstr(str)

OUTPUT:



```

C:\Users\Oell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\inglystr.py
Enter the string raising
raisingly

Process finished with exit code 0

```

RESULT:

Program to add “ly” and “ing” to a string has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:8

LONGEST WORD

Date: 14/12/2022

AIM: Accept a list of words and return length of longest word.

ALGORITHM:

Step 1: Start.

Step 2: Initialize an empty list and size of list.

Step 3: Using for loop access each element and add to list.

Step 4: Check the length of each element and compare with first element in list. If it is greater than first element then store in temp variable.

Step 5: Print the word.

Step 6: Stop.

PROGRAM CODE:

```
a=[]

n=int(input("Enter the size of the list "))

for i in range(1,n+1):

    print("Enter the ",i," element ")

    inp=input()

    a.append(inp)

maxl=len(a[0])

temp=a[0]

for i in a:

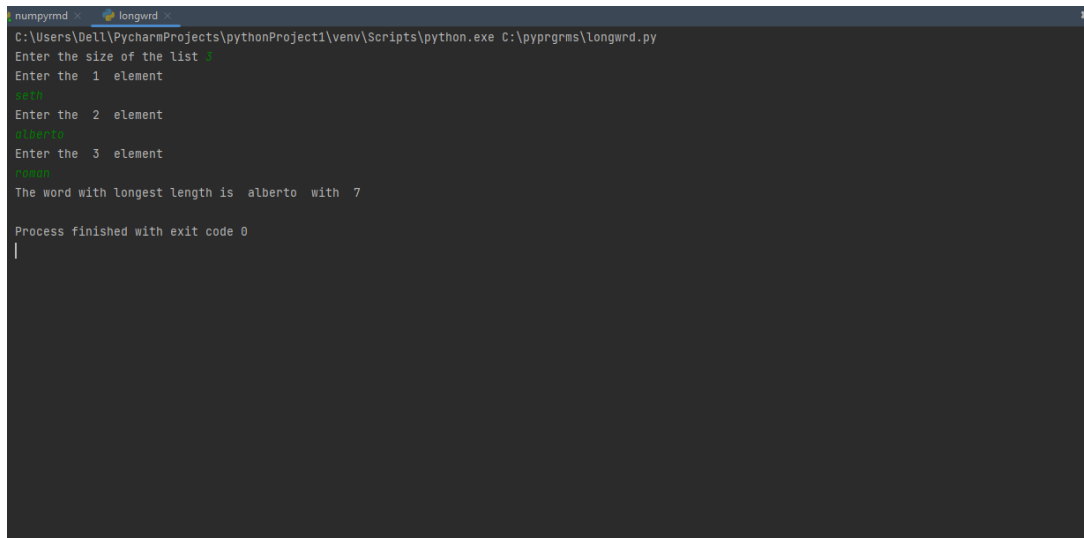
    if len(i)>maxl:
```

```
max1=len(i)
```

```
temp=i
```

```
print("The word with longest length is ",temp," with ",max1)
```

OUTPUT:



```
numpymd -x longwrd
C:\Users\Dell\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\pyprgrms\longwrd.py
Enter the size of the list :
Enter the 1 element
alberta
Enter the 2 element
alberta
Enter the 3 element
common
The word with longest length is  alberta  with  7
Process finished with exit code 0
|
```

RESULT:

Program to get longest word in a list has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:9

PRINT PATTERN

Date: 12/12/2022

AIM: Construct following pattern using nested loop.

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * *  
* * *  
* *  
*
```

ALGORITHM:

Step 1: Start.

Step 2: Input the maximum range of pattern.

Step 3: Use outer for loop and inner for loop and print the pattern till the range.

Step 4: Then reverse the range in step 4 and print the pattern.

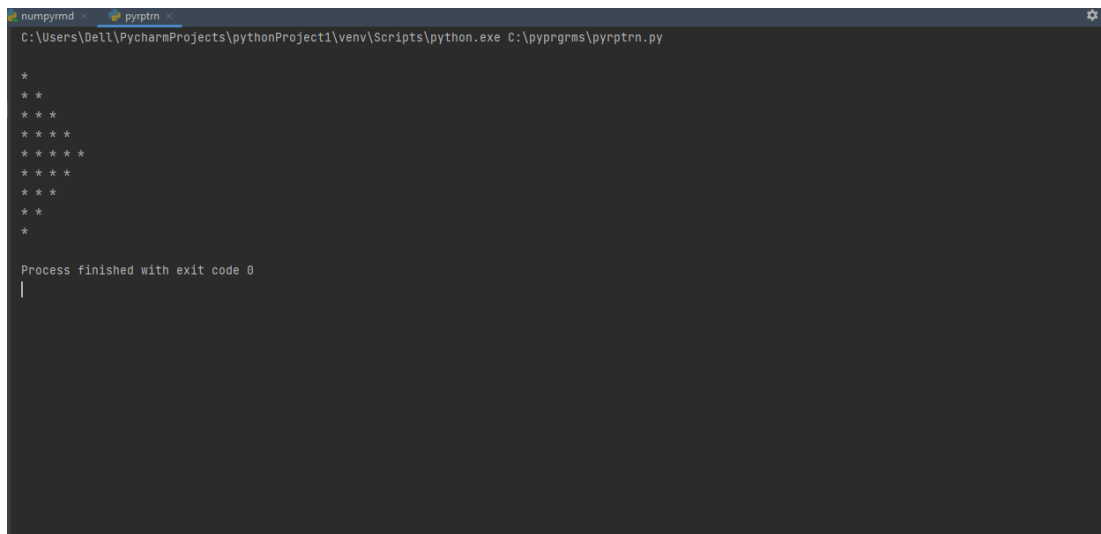
Step 5: Stop.

PROGRAM CODE:

```
n=5  
  
for i in range (0,n):  
    for j in range(0,i):  
        print("*",end=" ")  
  
    print(" ")
```

```
for i in range(n,0,-1):  
    for j in range(0,i):  
        print("*", end=" ")  
    print(" ")
```

OUTPUT:



The screenshot shows a terminal window with the following output:

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * *  
* * * *  
* * *  
* *  
*  
  
Process finished with exit code 0  
|
```

RESULT:

Program to print pattern has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:10

PRINT PATTERN

Date: 19/12/2022

AIM: Generate all factors of a number.

ALGORITHM:

Step 1: Start.

Step 2: Input a number to find the factor.

Step 3: Check the divisibility of number in range 1 till number.

Step 4: Print if number divisible by i.

Step 5: Stop.

PROGRAM CODE:

```
num=int(input("Enter the number to find factor "))
```

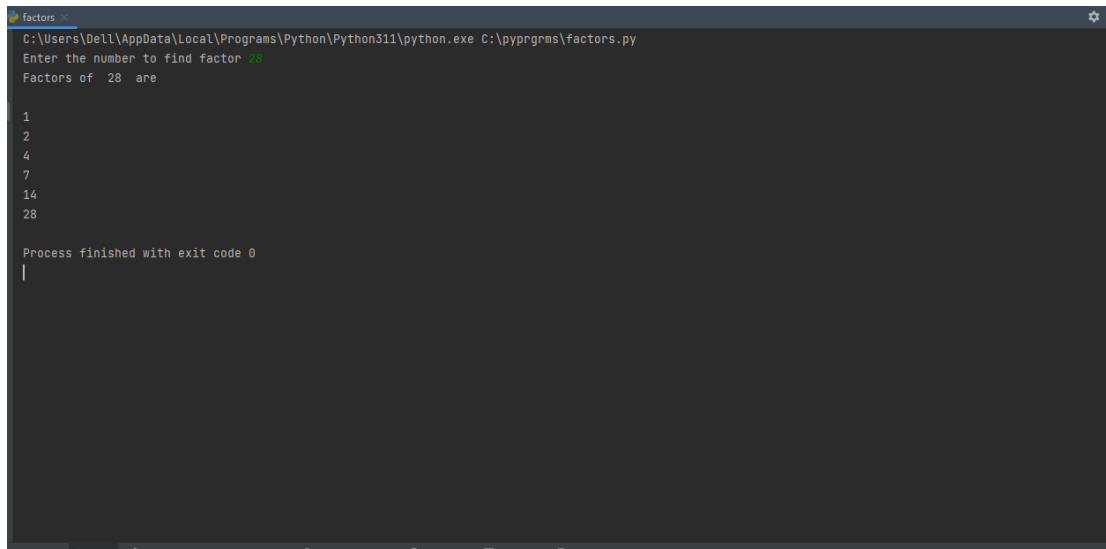
```
print("Factors of ",num," are\n")
```

```
for i in range(1,num+1):
```

```
    if num%i==0:
```

```
        print(i)
```

OUTPUT:



```
factors
C:\Users\Dell\AppData\Local\Programs\Python\Python311\python.exe C:\pyprgrms\factors.py
Enter the number to find factor 28
Factors of 28 are
1
2
4
7
14
28

Process finished with exit code 0
|
```

RESULT:

Program to find factors has been executed successfully and output is verified.

LABCYCLE 2

EXPERIMENT NO:11

LAMBDA FUNCTIONS

Date: 19/12/2022

AIM: Write lambda functions to find area of square, rectangle and triangle.

ALGORITHM:

Step 1: Start.

Step 2: Initialize a function with lambda function with arguments for square, rectangle and triangle.

Step 3: Input the dimensions for square, rectangle and triangle.

Step 4: Call each 3 three functions and print values.

Step 5: Stop.

PROGRAM CODE:

```
area_s=lambda a : a*a

area_rect=lambda l,b : l*b

area_triangle=lambda b1,h :0.5*b1*h

a=int(input("Enter the side of the square "))

print("Area of square ",area_s(a))

l=int(input("Enter the length of rectangle "))

b=int(input("Enter the breadth of rectangle "))

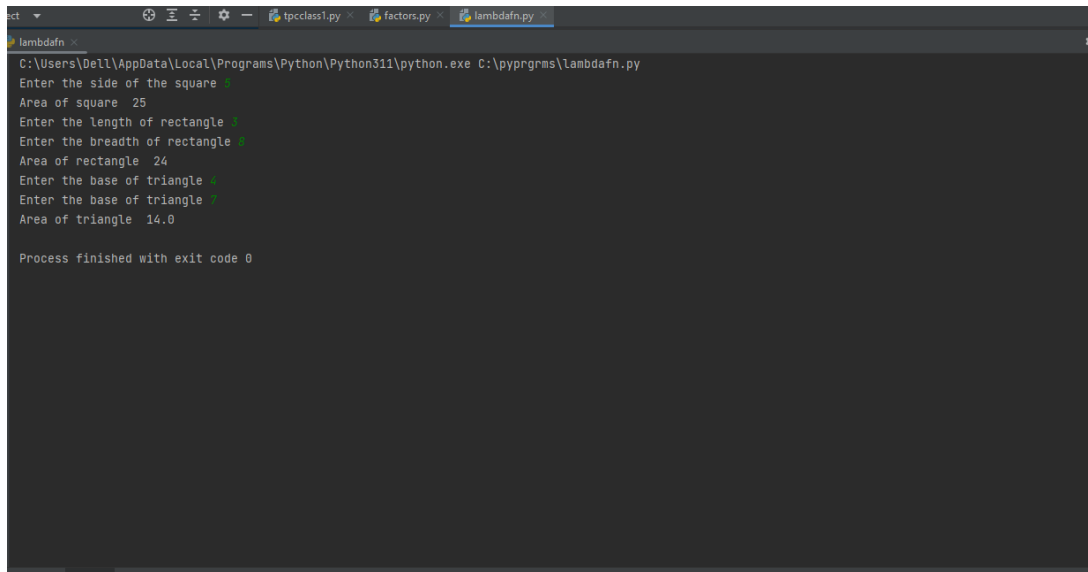
print("Area of rectangle ",area_rect(l,b))

b1=int(input("Enter the base of triangle "))

h=int(input("Enter the base of triangle "))

print("Area of triangle ",area_triangle(b1,h))
```

OUTPUT:



```
ct ▾
tpcclass1.py factors.py lambdafn.py
lambdafn
C:\Users\Dell\AppData\Local\Programs\Python\Python311\python.exe C:\pyprgrms\lambdafn.py
Enter the side of the square 5
Area of square 25
Enter the length of rectangle 3
Enter the breadth of rectangle 8
Area of rectangle 24
Enter the base of triangle 7
Enter the height of triangle 4
Area of triangle 14.0

Process finished with exit code 0
```

RESULT:

Program to perform lambda function has been executed successfully and output is verified.