Program No: 1

Aim: Square of n numbers in Python

```
for n in [0, 1, 2, 3, 4]
    square = n**2
    print('n, I squared is', square)
print("The for loop is completed!")
```

Result: The program has been executed and output is verified

output

0 squared is 0
the for loop is completed
1 squared is 1
the for loop is completed
2 squared is 4
The for loop is completed
3 squared is 9
The for loop is completed
4 squared is 16
The for loop is completed

Program no: 2

Aim: Program a list of vowels selected from given word

```
str1 = input ("Enter a string:")
str1_lower = str1.lower()
vowels = "aeiou"
count = 0
for i in str1_lower:
    count = count + 1
print()
print ("count of vowels in the given string:",
    count)
```

Result:

The program has been executed and output is verified

output

enter a string: ammy

a

count of vowels in the given string: 1

a

count of vowels in the given string: 2

Program NO: 3

Aim: Count the occurrences of each word in a line of text

```
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1
    return counts

print(word_count('I love my India !'))
```

Result:

The program has been executed and output is verified.

OUTPUT

{'a': 1, 'love': 1, 'my': 1, 'data': 1}

Program No: 4

Aim: Store list of first name. count the occurrence of ('a') within the list

```
text_str = "hai ammus"
count = 0
for i in text_str:
if i == 'a';
count = count + 1
print ("count of a in hai ammus is :" + str(count))
```

Result: The program has been executed and output is verified

Output

Count of a in hai ammu is : 2

Program No: 6

Aim: Enter 2 lists of integers check

a) whether list are of same length

b) whether list sums to same value

c) whether any value occur in both output.

```
def list():
list1=[]
list2 =[]
list3=[]
n1= int (input ("total number of elements in to list 1:"))
for i in range (n1):
 val= int (input ("enter a number":))
list1.append(val)
n2= int(input ("total number of element to list 2:"))
for i in range(n1):
val = int (input ("enter a number":))
list.append (val)
if(n1==n2):
   print ("list are of same length")
 else:
   print ("list are not of same length")
list3 = [each for each in list1 if each in list2 ]
print ("values in the both list are :", list3)
list()
```

output

total number of element in list 1: 4

enter a number: 5

enter a number: 6

enter a number: 2

enter a number: 4

total number of element in the list: 5

enter a number: 2

enter a number: 5

enter a number: 7

enter a number: 8

enter a number: 9

list are not same length:

sum value is not same

values in the both lists are: [5, 2]

Program No: 6

Aim: get a string from an input string change occurrences of first character replaced with $ except first character

```
def change_char[str1]:
char =str1[0]
str1 = str1.replace(char,'$')
str1 = char +str1[1]
return str1
print(change_char("ammoras"))
```

Result: the program has been executed and the output is verified.

Output

ammca$s

Program No: 7

Aim: Create a string from given string where first and last characters exchanged.

```
str = input ("Enter a string :")
new_str = str[-1:] + str[4:-1] + str[:1]
print (new_str)
```

Result: The program has been executed and output is verified

OUTPUT

Enter a string: hai

lah

Program no: 8

Aim: Accept the radius from user and find area of circle

```
import math
r = float(input("Enter the radius of circle:"))
area = math.pi * r * r
print("%.2f" %area)
```

Result: The program has been executed and output is received

OUTPUT

Enter the radius of the circle: 5

78.54

Program No: 9

Aim: Accept an integer n and compute n+nn+nnn

```
n = int(input("enter a number n:"))
temp = str(n)
t1 = temp + temp
t2 = temp + temp + temp
comp = n + int(t1) + int(t2)
print("the value is:", comp)
```

Result: the program has been executed and output is verified

output

enter a number a : 1

The value is : 123

Program NO:10

Aim: Sort dictionary in ascending and desending order

```
import operator
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
print("ascending order:",s)
s1 = dict(sorted(d.items(), key = operator.itemgetter(1),
reverse = True))
print("descending order:",s1)
```

Result: The program has been executed and output verified

output

dictionary: {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}

ascending order: [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]

descending order: {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}

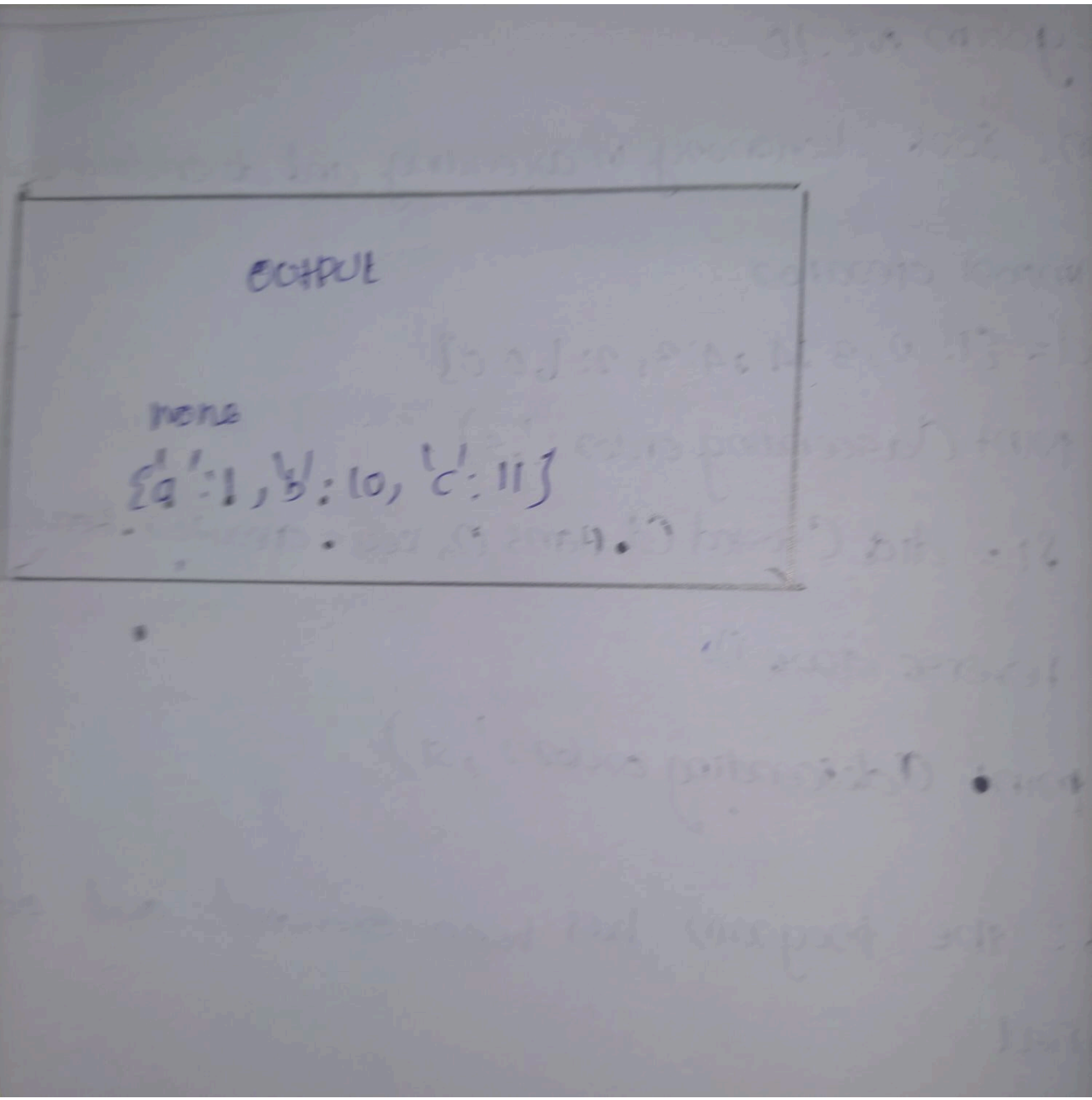
Program No: 11

Aim: merge two dictionaries

```
x = {'a': 1, 'b': 2}
y = {'b': 10, 'c': 11}
z = x.update(y)

print(z)
print(x)
```

Result: The program has been executed and output is verified.


OUTPUT
mena
{'a':1,'b':10,'c':11}

Program NO: 12

Aim: Find gcd of 2 numbers

```
num1 = int(input("Enter 1st number:"))
num2 = int(input("Enter 2nd number:"))
i=1
while(i<=num1 and i<=num2):
    if(num1%i==0 and num2%i==0):
        i=i+1
while(i<=num1 and i<=num2):
    if(num1%i==0 and num2%i==0)
        gcd=i
        i=i+1
print("GCD is", gcd)
```

Result: the program has been executed and output is verified.

Output

Enter 1st number: 12

enter 2nd number: 6

GCD is 6.

Aim: Write a program to find biggest three numbers entered by the user

```
num1 = float(input("enter first number:"))
num2 = float(input("enter second number:"))
num3 = float(input("enter third number:"))
if (num1 > num2) and (num1 > num3):
largest = num1
elif (num2 > num1) and (num2 > num3):
largest = num2
else:
largest = num3
print("the largest number is", largest)
```

Results: The program has been executed and output is verified.

output

Enter first number: 1
enter second number: 2
enter third number: 8
The largest number is 8.0

program no:14

Aim: write a python program to find a square of number entered by the user

```
number = float(input("please enter any numeric value :"))

square = number * number

print("the square of given number {0} = {1}".
format(number, square))
```

Result: the program has been executed and output is verified.

Output

Please enter any numeric value : 5

The square of given number 5.0 = 25.0

Program no: 15

Aim: create a python program to return area of circle using function

```
import math
r=float(input("enter the radius of the circle:"))
area = math.pi * r*r.
print("area:",area)
```

Result: the program has been executed and output is verified.

Output

Enter the radius of the circle : 5

78.54.

Program NO: 16

Aim: From a list of integers, create a list removing an even numbers

```
list = [11, 22, 33, 44, 55, 66]
print ("original list")
print (list)
for in list:
    if (i%2 == 0):
        list.remove(i)
print ("list after removing an even numbers:")
print (list)
```

Result: The program has been executed and output is verified.

Program No: 16

Aim: From a list of integers, create a list removing an even numbers

```
list = [11, 22, 33, 44, 55, 66]
print ("original list")
print (list)
for in list:
    if (i%2 == 0):
        list.remove(i)
        print ("list after removing an even numbers:")
        print (list)
```

Result: The program has been executed and output is verified.

OUTPUT

original list

[11, 22, 33, 44, 55, 66]

list after removing an even numbers:

[11, 33, 44, 55, 66]

list after removing an even numbers:

[11, 33, 55, 66]

list after removing an even numbers:

[11, 33, 55]

Program No: 17

Aim: program to find the factorial of a number

```
def factorial(n):
return 1 if (n==1 or n==0) else n*factorial(n-1);
num=5;
print("factorial of", num, "is", factorial(num))
#
```

Result: The program has been executed and output is verified

OUTPUT

Factorial of 5 is 120

program 180:18

Aim: Generate Fibonacci series of N terms

```
nterms = int(input("how many terms?"))
n1, n2 = 0, 1
count = 0
if nterm <=0:
print("please enter a positive integer")
elif nterms == 1:
print("Fibonacci sequence upto", nterms, ":")
print(n1)
else:
print("Fibonacci sequence :")
while count <nterms:
print(n1)
nth = n1+n2
n1=n2
n2 =nth
count += 1
```

Output

How many terms? 10

Fibonacci Sequence:

0
1
1
2
3
5
8
13
21

Program no: 14

Aim: Find the sum of all terms in a list

```
list = [ ]
num = int (input ("How many numbers:"))
for n in range(num):
    numbers = int (input ("Enter a number"))
    list.append (numbers)
print ("sum of elements in given list is :", sum(list))
```

Result: the program has been executed and output is verified

Output

How many numbers : 5

Enter number 5

enter number 10

enter number 15

enter number 25

enter number 30

Sum of elements in given list is : 85

Program No: 20

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

```
def call():
    n=0
for x in range (1000, 10000):
num = str(x)
number = int(x)
first = int(num[0])
second = int(num[1])
third = int(num[2])
fourth = int(num[3])
if first%2 == 0:
if second%2==0:
if fourth%2==0:
for i in range(2, number):
if i*i == number:
print(number)
call()
```

Result: The program has been executed and output is verified.

Output

4624

6084

6400

8464

Program No: 21

Aim: display the given pyramid with step number accepted from user. Eg N=4

1

2 4

3 6 9

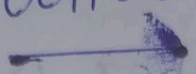
4 8 12 16

```
def pyr():
n= int(input("enter the number:"))
i =1
for i in range(1,n+1):
  j=1
for  j in range(1,i+1):
temp = i*j;
print(temp, end =" ")

pyr()
```

Result: the program has been executed and output is verified

Output



Enter the number : 4

1

2 4

3 6 9

4 8 12 16

Program No: 22

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

```
def char_frequency(str1):
dict = {}
for n in str1:
keys = dict.keys()
if n in keys:
dict[n] += 1
else:
dict[n] = 1
return dict
print(char_frequency('google.com'))
```

Result: The program has been executed and output is verified

Output

{'g': 2, 'o': 3, 'd': 1, 'e': 1, 'n': 1, 'c': 1, 'm': 1}

Program No - 23

Aim: Add 'ing' at the end of a given string: If it already ends with 'ing', then add 'ly'

```
def add_string(str1):
length = len(str1)
if length > 2:
if str1[-3:] == 'ing':
str1 += 'ly'
else:
str1 += "ing"
return str1
print(add_string('ab'))
print(add_string('abc'))
print(add_string('string'))
```

Result: The program has been executed and output is verified

OUTPUT

ab

abang

strongly

Program NO: 24

Aim: Accept a list of words and return length of longest word

```
def find_longest_word(words_list):
word_len = []
for n in words_list:
  word_len.append((len(n),n))
  word_len.sort()
  return word_len[-1][0],word_len[-1][1]
result = find_longest_word(["php","exercises","Python"])
print("Longest word:",result[1])
print("Length of the longest word:",result[0])
```

Result:

The program has been executed and output is verified

OUTPUT

Longest word : python

length of the longest word : 7.

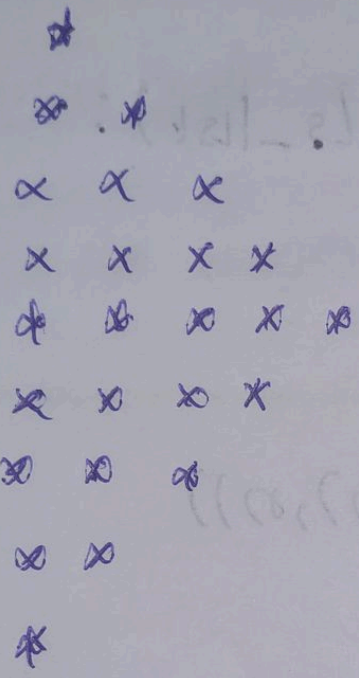
Program No: 25

Aim: Construct following pattern using nested loop

```
n=6;
for i in range (n):
for j in range (i):
print ('x', end="")
print ()
for i in range (n, 0, -1):
for j in range(i):
print ('x', end="")
print ()
```

Result: The program has been executed and output is verified.

остров



Program No: 26

Aim: Generate all factors of a number

```
def print_factors (x):
print ("the factors of", x, "are:")
for i in range (1, x+1):
if x % i == 0
print (i)
num = 320
print_factors (num)
```

Result: the program has been executed and output

verified

output

The factorial of 320 are :

1
2
4
5
8
10
16
20
32
40
64
80
160
320

Program No: 27

Aim: write lambda functions to find area of square, rectangle and triangle.

```
import math

t_perl = lambda p,q,r : p+q+r
s_area = lambda len, bt : len * bt
c_perl = lambda rad : 2*math.pi *rad
c_area = lambda rad : math.pi *rad*rad

print ("perimeter of triangle (10,20,15) is :", t_perl
(10,20,15))
print ("area of rectangle (30,20) is :", s_area (30,20))
```

Result: the program has been executed and output is verified

Output

Perimeter of Triangle (10, 20, 15) is: 45

Area of Rectangle (30, 20) is: 600