

Programs No:1

Aim : Python program to find area.

```
def area(r):
```

```
Pi = 3.14
```

```
return Pi * (r * r);
```

```
num = float (input("enter the value for :")).
```

```
Print ("Area is %o.6f" % area (num)).
```

Result :

The program has been executed
and the output was verified.

Output:

Enter the value for : 3.

Area is 28.26000.

Program No: 2

Aim: Python programs to find largest among 3 Numbers.

number 1 = float (input ("enter the first number"))
number 2 = float (input ("enter the second number"))
number 3 = float (input ("enter the third number"))

If (number1 > number2) and (number1 > number3)
 largest = number 1.

elif (number2 > number1) and (number2 > number3)
 largest = number 2.

else :

 largest = number3

Print ("The largest number is ", largest).

Result:

The programs has been executed and output was verified.

Output

Enter the first number : 2

Enter the second number : 4

Enter the third number : 5

The largest number is 5

Program No :3.

Aim : Python programs to find square of a number

digit = int(input("Enter an integer number:"))

square = digit * digit

printf("Square of %d is %d"),

Result :

The program has been executed and output was verified.

Output

Enter an integer number : 4.

Square of 4 is 16.

Program No:4

Aim : Python programs to find area of Circle

From math import pi.

r = float (Input the radius of the circle : ")

Print ("the area of the circle with rad
+ str(r) + " is :" + str (pi * r ** 2))

Result:

The program has been executed and
output was verified

Output

Input the radius of Circle: 4.

The area of the circle with radius
50.2654

Program No : 5

Aim: Python programs to find square of n.

List 1 = [14, 20, 13, 8, 6, 2].

for n in list 1:

Square = n * n.

Print (n, squared is ', square).

Result.

The program has been executed and the output was verified.

Output.

14 squared is 196.

20 squared is 400

13 squared is 169.

8 squared is 64

6 " Squared is 36

2 squared is 4.

Program :- b

Aim: Python program to find vowels in a string.

String A = "Hello .. How are you"

Print ("Given string : \n", stringA).

Vowels = "AaEeIiOoUu".

res = Set (each for each in string A if each in Vowels)

Print ("The vowels present in the string :\n", res).

Result:

The program has been executed and output was verified

Output

Given string :

Hello... how are you!

The vowels in the string :

{'u', 'a', 'e', 'o'}

Program 7

Aim : Python programs to count words in a

def wordCount(str):..

Counts = dict()

Words = str.split()

for word in words:

if word in counts:

Counts[word] += 1

else:

Counts[word] = 1.

returns Counts.

Print(wordCount('When you change the quality of thinking, you change the quality of your life sometimes instantly')).

Result:

The program has been executed and output was verified.

Output:

```
{'when': 1, 'you': 2, 'change': 2, 'the': 2, 'quality': 2,  
'of': 2, 'gone': 2, 'thinking': 1, 'life': 1, 'sometimes':  
'instantly': 1}.
```

Program No: 8.

Ans: Python program to count a in a list.

a = ['Anto', 'Rehan', 'Billan', 'Doris'].

str1 = (''.join(a)).

Count = 0.

for i in str1:

if i == 'a':

Count = Count + 1.

Print("Count of a in the list is:"

+ str(count)).

Result:

The program has been executed and the result was verified.

Output

Count of a in the list is 3.

Program No:9.

Aim: Python program to check the lengths.

List 1 = [10, 10, 11, 12, 13, 14, 16, 15, 16, 12].

List 2 = [16, 12, 13, 14, 15, 16, 10, 11, 12, 10, 12].

len1 = len(list1).

len2 = len(list2)

If len1 = len2:

Print ('both list have equal lengths').

else:

Print ('both list doesn't have equal lengths').

Result.

The program has been executed and verified.

Output

Both list have equal length.

Program No. : 10

Aim: Python programs to check the sum of lists.

list 1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 10]

list 2 = [16, 12, 13, 14, 15, 16, 10, 11, 13, 10, 15]

total 1 = sum (list 1).

total 2 = sum (list 2).

If total 1 == total 2 :

Print ('both list have equal sum').

else :

Print ('both list doesn't have equal sum')

Result

The program has been executed and output was verified.

Output:

Both list have equal sum.

Program No: 11

Aim: Python programs to check the common elements in the lists.

list 1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 10]

list 2 = [10, 10, 11, 12, 12, 16, 14, 16, 15, 19, 12].

for value in list 1:

 if value in list 2:

 Common = 1.

 if Common == 1:

 Point ("there are common elements").

else: print ("no common elements").

Result!

The program has been executed
output was verified.

Output

There are common elements.

Program No:12.

Aim: Python programs to replace a character

```
def change - char (str1):  
    char = str1 [0].  
    str1 = str1.replace (char, '$').  
    str1 = char + str1 [1: ].  
    print (change - char ('refresh')).
```

Result:

The program has been executed and output was verified

Output

ref \$esh

Program No : 13.

Sim : Python programs to exchange the
and last letter in string

```
def change_string(str1):
```

```
    return str1[-1:] + str1[1:-1] + str1[:1]
```

```
Print(change_string('pineapple')).
```

Result

The program has been executed
output was verified.

Output:

eineapplp

Program No : 14.

Aim : Python programs to merge 2 dictionaries

def merge (dict1,dict2):

 returns (dict2 . update (dict1)).

dict 1 = { 'a': 10, 'b': 8 }.

dict 2 = { 'c': 5, 'd': 2 }

Print (Merge (dict1,dict2)).

Print (dict2).

Result :

The program has been executed and was verified.

Output

None

{'d':5, 'c':2, 'a':10, 'b':8}.

Program : 15

Aim : Python program to ascend descend dict

Import Operator

```
d = {1: 0, 3: 4, 4: 3, 2: 1, 0: 0}
```

```
Print ('Original dictionary : ', d).
```

```
Sorted -d = Sorted (d.items(), Key = Operator.itemgetter)
```

```
Print ('Dictionary in ascending order by value : ',  
      sorted -d ).
```

```
Sorted -d = dict(Sorted (d.items(), Key = Operator.itemgetter,  
                      reverse = True)).
```

```
Print ('Dictionary in descending order by value : ',
```

Result

The program has been executed and output was verified.

Output

Original dictionary : $\{1:2, 3:4, 4:3, 2:1, 0:0\}$.

Dictionary in ascending order by value :

$\{(0,0), (2,1), (1,2), (4,3), (3,4)\}$.

Dictionary in descending order by value :

$\{3:4, 4:3, 1:2, 2:1, 0:0\}$.

Program 16

Aims : Python programs to remove even numbers from list.

list = (11, 22, 33, 44, 55, 66, 77, 88, 99).

Print (list).

```
for i in list:  
    if (i % 2 == 0)  
        list.remove(i)
```

Print ("List after removing : ", list).

Result

The program has been executed
output was verified.

Output: [11, 22, 33, 44, 55, 66, 77, 88, 99].

List after removing: [11, 33, 55, 77, 99]

Program No : 17

Aim: Python programme to find gcd of two

def gcd (a,b) :

If ($b == 0$)

return a.

return gcd (b, a % b).

a = 45

b = 65

If (gcd(a,b)) :

Print ('gcd of ', a, 'and ', b, 'is', gcd (a,b))

else :

Print ('not found')

Result

The program has been executed a
was needed.

Output

GCD of 45 and 65 is 5.

(1) Euclidean algorithm
for GCD of 45 and 65

(1)

: to

(0 = 0.0)

(1) minor list

(total & previous value +

longitudes need not change
between steps

Program No:18:

Aim: Python programs to find factorial of a number

```
num = int (input ("enter a number:")) .
```

```
factorial = 1
```

```
if num < 0 :
```

```
    print ("Sorry, factorial does not exist for negative  
          numbers") .
```

```
elif num == 0:
```

```
    print ("The factorial of 0 is 1") .
```

```
else :
```

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

```
        factorial = factorial * i
```

```
    print ("The factorial of ", num, "is", factorial)
```

Result.

The program has been executed and the output was verified.

Output

Enter a number: 5

The factorial of 5 is 120.

(d.s.A.d)

(a.a) bsp. 211 of bsp. p { }

(Chowdhury)

Program No : 19

Aim: Python program to find fibonacci sequence.

```
def recure-fibo(n):
```

```
    if n <= 1:
```

```
        return n
```

```
    else:
```

```
        return (recure-fibo(n-1) + recure-fibo (n-2))
```

```
nterms = int(input("How many terms?"))
```

```
If n terms <= 0:
```

```
    print ("Please enter a positive integer").
```

```
else:
```

```
    print ("Fibonacci sequence:").
```

```
    for i in range(0, terms):
```

```
        print (recure-fibo(i)).
```

Result

The programs has been executed and the output was verified.

Output.

How many deoms ? . 4.

Fibonacci sequence :

0

1

1

2.