

PROGRAM - I

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 %.6f" %area(num))

Result :

The program has been executed and
the output was verified.

Output

Enter the value for : 3

Area is 28.2600

Program No: 2

AIM: Python Program 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 (Number 1 > Number 2) and (Number 1 > Number 3):
    largest = Number 1
elif (Number 2 > Number 1) and (Number 2 > Number 3):
    largest = Number 2
else:
    largest = Number 3

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

Result:

The program has been executed and
the 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 ends here.)

(Execution ends.)

Two factors used and mapped.

Biggest came top.

PROGRAM : 3

AIM : Python programs to find square
of a number

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

```
Print(f " Square of {digit} is {square}")
```

Result :

The program has been executed and
output was verified

Output is read from a composite file

Enter an integer number : 4

Square of 4 is (6)

Program : 4

AIM: Python program to find area of circle

```
From math import pi
```

```
r=float(input("Enter the radius of the circle :"))
```

```
print("The area of the circle with radius",
```

```
+str(r)+" is : "+str(pi*r**2))
```

Result :

The program has been executed and
the output was verified

Output ~~if it is correct work~~
~~random~~

Input the radius of the circle : 4

The area of the circle with radius 4
is 50.2654

Example is 50.2654 example ?

for between and majorly
between and

PROGRAM : 5

AIM: Python program do find Square of "n" numbers present in list.

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 Square is 196

20 Square is 400

13 Square is 169

8 Square is 64

6 Square is 36

2 Square is 4

PROGRAM : 6

AIM: Python Program to find vowels in a String

String A = "Hello... how are you"

Print ("Given String :\n", string1)

vowels "AaEeIiOoUu"

res = Set ([each for each in string1
if each in vowels])

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

Result :

The program has been executed and the output was verified.

Output

Given string:

Hello. How are you? All?

The vowels present in the string:

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

Example: it belongs to the

PROGRAM - 7

AIM : Python programs do count words in a sentence

```
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('life is a stream  
of memory'))
```

Result :

The program has been executed and the output was verified

Output

```
{'life': 1, 'is': 2, 'a': 3, 'the': 4,  
'of': 5, 'memory': 6}
```

Output 2

Comparison word count for
("possible", "at", "probably", "newest") is

"possible" > "at" > "probably" > "newest"

Arguable all these words have been
("possible", "at", "probably", "newest")

Count

verb has been used more than once in the sentence
("possible", "at", "probably", "newest")

PROGRAM - 8

AIM: Python Programs to count a in a list
It finds how many times a is there.

```
a = ['anu', 'aku', 'aashi', 'appu']
```

```
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 output was verified

Output

Count of a in the list is : 5

Programs - 9

AIM : Python Programs do check the length of lists

```
list1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 12]
```

```
list2 = [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 length')
```

```
else:
```

```
    print('both list does not have equal  
length')
```

Result :

The program has been executed and output was verified

Output

both list have equal length.

Program which takes two

(class . 11)

Output

is like

of list

(function form)

which will go down.

(class 11)

but following will

and working of

PROGRAM - 10

AIM : Python Programs to check the sum of lists

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

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

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 have equal sum')

Result :

The program has been executed and the output was verified.

output

Both kid have equal sum

(A) kid can

(C) kid can

→ good

→ it's good to play and have fun

long word for each kid → good

→ it's good

PROGRAM - 11

AIM : Python Program to check the common element in the lists

list1 = [10, 10, 11, 12, 12, 13, 14, 16, 15, 16, 12]

list2 = [10, 10, 11, 12, 12, 16, 14, 16, 14, 16, 15, 19, 12]

for value in list1:

if value in list2:

common = 1

if common == 1:

Print ("There are common elements")

else:

Print ("no common elements")

Result :

The program has been executed and the output was verified.

01-07-2019

Output.

There are common elements

(Giant) Crows

(Giant) Crows

(Giant) Crows

(Crows kept near the abode)

(Crows kept near the abode)

PROGRAM - 12

AIM : Python Program to replace a character

def change_char(sdh):

char = sdh[0]

str1 = sdh.replace(char, '\$')

sdh = char + str1[1:]

Print(change_char('refresh'))

Result :

The program has been executed and the output was verified.

Output

refresher

[51, 51, 51, 51, 51, 51, 51, 51, 51]

[51, 51, 51, 51, 51, 51, 51, 51, 51]

field with pulse

51 51 51 51

51 51 51 51

51 51 51 51

(channel number was used "51")

(channel number on the

PROGRAM : 13

AIM : Python programs do exchange the first and last letter in a string

def change_string(sstr):

return sstr[-1:] + sstr[1:-1] + sstr[:1]

Print(change_string('pineapple'))

Result :

The program has been executed and output was verified

Output

eine applicability analysis

(Q18) node spread

Definition

((ϕ , node)) \rightarrow $\phi \in \text{nodes}$

End node + node

((λ node)) node spread

also can follow the same kind representation

basically same

PROGRAM - 14

AIM: Python program to merge 2 dictionaries

```
def Merge(dict1, dict2):
```

```
    dict2.update(dict1)
```

```
dict1 = {'a': 10, 'b': 8}
```

```
dict2 = {'d': 5, 'c': 2}
```

```
Print(Merge(dict1, dict2))
```

```
Print(dict2)
```

Result:

The program has been executed and output was verified.

Output

Name

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

[1:7]mb+[7-11]mb+[11-14]mb

(0:stg+0:9) mb+(0:8)mb

target for disksize and offset

PROGRAM : 15

AIM: Python Program to ascend and descent dictionary
Input Operator

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

Print ('Original dictionary ::', d)

Sorted - d = sorted (d.items(), key = operator.
itemgetter(1))

Print ('Dictionary in ascending order by value ::',
Sorted - d)

Sorted - d = dict (sorted (d.items(), key = operator.
itemgetter(1), reverse = True))

Print ('Dictionary in descending order by value ::',
Sorted - d)

Result:

The program has been executed and the
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:

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

(Sort (dict) against key)

(Sort by key)

Sort between dict and map with
holding key and value

PROGRAM : 16

AIM: Python program do remove even numbers from the 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 and
the output was verified

3.1.1. MAGNA

Output

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

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

robbergo weg

{11, 33, 55, 77, 99}

(11, 33, 55, 77, 99)

and/or -eq -gt (or eq, gt) followed by . b. b. b.

(11, 33, 55, 77, 99)

and/or -eq -gt (or eq, gt) followed by . b. b. b.

(11, 33, 55, 77, 99)

and/or -eq -gt (or eq, gt) followed by . b. b. b.

(11, 33, 55, 77, 99)

and/or -eq -gt (or eq, gt) followed by . b. b. b.

(11, 33, 55, 77, 99)

(11, 33, 55, 77, 99)

PROGRAM - 17

AIM: Python Program to find gcd of number

def gcd(a,b):

if (b == 0):

return a

return gcd(b, a % b)

a = 45

b = 65

if (gcd(a,b)): else if (gcd(a,b) == 0):

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

else:

Print ('not found')

Result :

The program has been executed and the output was verified

SI : MASTERS

Output

GCD of 45 and 65 is 5

[PE, 38, FF, 3d, 81, 8A, 8E, 52, 41]

(Ans) 5

Find GCD of 45 and 65

(Ans) 5

(Ans) 5

(Ans) 5

PROGRAM - 18

AIM: Python program 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 number")

```
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

Endreka keemusid: 5! = 120

The factorial of 5 is 120

(d, e)

(c, d)

(a)

(d, e, b, c) bog

((d, e))

((d, e)) bog! Endreka keemusid: 4!

PROGRAM - 19

English

AIM: Python program to find fibonacci sequence

def recer-fibo(n):

positive

if n <= 1:

return n

else

return (recer-fibo(n-1) + recer-fibo(n-2))

ndterms = int(input("How many terms? ."))

if ndterms <= 0:

Print ("please enter a positive integer")

else:

Print ("Fibonacci Sequence : ")

for i in range(ndterms):

Print (recer-fibo(i))

Result :

The program has been executed and the output was verified

output

How many terms? 4

Fibonacci Sequence:

0

1

1

2

first term took book knowledge, fence

children will

use it

class to learn book with it

class to learn book with it

class to learn book with it

PROGRAM - 20

AIM : Python program to perform string function.

```
def add_string(sstr):
    length = len(sstr)
    if length > 1:
        if sstr[-3:] == "ing":
            sstr += 'ly'
        else:
            sstr += 'ing'
    return sstr
```

```
Print (add_string ('do'))
```

```
Print (add_string ('according'))
```

Result :

The program has been executed and the output was verified.

Output

done if file ab mapped

according

(0) odd

(read from 0 to 7-odd-number)

(0, even) from back to front

(read from 0 to 7-odd-number)