

## Programs 1

Aim: Write a python program to find the sequence of the number entered by the user?

### Program

```
num = float(input("Please enter your value"))
Square = num * num
print("The square of the given value is {} = {}".format(num, Square))
```

### Result

The program has been executed and output is verified.

Output

Please enter your value 5

The square of the given value is  $5.0 = 25.0$

## Program 2

High

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

### Program

```
def findarea(r):  
    PI = 3.142  
    return PI*(r*r)  
num = float(input("Enter r value:"))  
print("Area is %.6f" % findarea(num));
```

### Result

The program is been executed and output is verified.

## Output

Scraped soft soil at non-piping condition in ditch bank  
Scrape soft soil from ditch, roadside soft soil

Enter or value : 83

Area is 216.452380 sqm(9'') larger than area

of different size having soft soil percentage > 5% (large  
(damp), medium, soft)

buffer bar between road and piping soil

buffer is

## Program 3

Page 3

Aim: Write a python program to find the biggest of the 3 numbers entered by the user?

### Program

```
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)
```

### Result

The program has been executed and output is verified.

So Output

## Output

for more number compare nothing is stored in  
variable so program shows 0

Enters the first number : 5

Enters Second number : 9

Enters third number : 7

The largest number is 9.000000000000001  
(After a certain place) due to round off error  
(Computation of 9.000000000000001 to 9.000000000000001)

so before less between need of compare will  
be different

## Program 4

Aim : List comprehension square of N numbers.

Program

```
[0, 1, 4, 9, 16, 25]
```

```
for x in range(100):  
    print(x**2)
```

Result

The program has been executed and output is verified.

# Output

10	784	386	7056
1	841	3249	707223
4	900	3364	7396
9	961	3481	7569
16	1024	3600	7744
25	1089	3721	7921
36	1156	3844	8100
49	1225	3969	8281
64	1296	4096	8464
81	1369	4225	8649
100	1444	4356	8836
121	1521	4489	9025
144	1600	4624	9216
169	1681	4761	9406
196	1764	4900	9604
225	1849	5041	9801
256	1936	5184	
289	2025	5329	
324	2116	5476	
364	2209	5625	
400	2304	5776	
441	2401	5929	
484	2500	6084	
529	2601	6241	
576	2704	6400	
625	2804	6561	
676	2916	6724	
729	3025	6889	

## Program 5

Aim: from a list of vowels selected from a given word  
{Input, I: list, I: word, I: separates, I: output}

## Program

```
def check_vow(string, vowels):
    final = [each for each in string if each in vowels]
    print (len(final))
    print (final)
String = "Ashique p rāj"
Vowels = "AaEeIiOoUu"
check_vow(String, Vowels)
```

## Result

The program has been executed and output was verified.

Output

A mapping

relation - to for example pronunciation but not

5

[‘A’, ‘i’, ‘u’, ‘e’, ‘ə’]

mapping

(long) vowel or diphthong  
(short) vowel

map

to give basis between word mapping self

between all

## Program 6

Topic 6

Aim : Count the occurrence of each word in a line  
of text.

Code: every word separated by a space

## Program

```
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("Hello Ashique how are you"))
```

## Result

The program has been executed and output has verified.

## ① Output

Output

{'Hello': 1, 'ashique': 1, 'how': 1, 'are': 1, 'you': 1}

Output

:Cebuano, penkti, iscov, gbooks, fbs  
Cebuano, davao, penkti, iscov, gbooks, fbs  
("Cebu" or "Ilocano"  
"Davao" or "Bukidnon")  
"Ilocano, penkti, iscov, gbooks, fbs  
"Mindanao" - El nido  
:Cebuano, penkti, iscov, gbooks, fbs

Output

fuctions for buttons used in my project

function

## Program 7

fragments

Aim: Store a list of first names. Count the occurrence of 'a' within the list?

## Program

```
test_str = "Hey Ashique how are you"
count = 0
for i in test_str
    if i == 'a':
        count = count + 1
print("Count of a in " + test_str + " is : " + str(count))
```

## Result

The program has been executed and output is verified.

## Output

Send a message to someone with count of a in key ashique how are you is:  
fr of to

comprob

:([fr] flowers [to] fr)

(fr) fr = at will

(fr) flowers, etc = at random

intention or function with

12 flowers on flower if

1-12 [function] flowers

at random

1-12 [function] intentions

at random

[up down and upside down] flowers [down] from

floral

higher level intentions and corresponded

behavior and

## Program 8

Aim : Enter two list of integer. Check

- a) whether list are of same length
- b) whether list sums to same value
- c) whether any values occur in both.

## Program

```
def lists():
    list 1 = []
    list 2 = []
    list 3 = []
    n1 = int(input("total number of elements in list 1:"))
    for i in range(n1):
        val = int(input("Enter the number:"))
        list 1.append(val)
    n2 = int(input("total number of elements in list 2:"))
    for i in range(n2):
        val = int(input("Enter a number:"))
        list 2.append(val)
    if(n1 == n2):
        print("list are of same length")
    else:
        print("list are not same length")
    if(sum(list 1) == sum(list 2)):
        print("sum value is same")
    else:
        print("sum value is not same")
```

```
list3 = [each for each in list1 if each in list2]
print ("value in the both lists are :" list3)
#output
```

## Result

The program has been executed and output is verified.

## Output

>> lists()  
total number of elements in list 1:3  
enter a number : 1  
enter a number : 4  
enter a number : 5  
total number of elements in list 2:3  
enter a number : 7  
enter a number : 8  
enter a number : 9  
list are same length  
sum value is not same  
value in the both list are : [ ]

>> lists()

total number of elements in list 1:2  
enter a number : 8  
enter a number : 9  
total number of element in list 2:2  
enter a number : 9  
enter a number : 7  
list are of same length  
Sum value of is not same  
values in the both list are : [9]

## Program 9

length()

Aim: Get a string from an input string where all occurrence of first character replaced with '\$', except first character.

## Program

Output

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

## Result

The program has been executed and output is verified.

[asked me about if I had an idea what "droon" was for  
. ("droon" = something that's off or outside") from  
Output

am \$1 jyothi  
begin here to discuss and work on speech self  
- bairavji

## Program 10

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

e: shiv to vivish

### Program

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

### Result

The program has been executed and output is verified.

P. 30

## Output

The code prints the words in a list  
in the following order from the screen,  
Enter a string: Ashique

Ashique

: (at 1 words) . Space at  
[0] 1 word  
(1 word) at 1 words at 1  
[1] 1 word at 1 words  
1 word  
(Catastrophe) words 0 words ) long

After  
long two line buffers used and memory set  
to 0

## Program 11

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

### Program

Import math:

```
r = float(input("Enter the radius of circle"))
area = math.pi * r * r
print("%1.2f" % area)
```

### Result

The program has been executed and output is verified.

## Output

of program

self made parts are craft parts or standard  
commercially available tool bus for

Enter the radius of circle : 5

78.54

(parts are not 3") larger  
Circle [1.1732 + 6.1732 = 42.00  
(42 - 20) from

two base bushes need not impinge at  
bottom

## Program 12

Intuition

Aim: Accept an integer  $n$  and compute  $n + n + nn$   
for  $n \in \mathbb{N}$ , i.e., suff: procedure

Program:

```
def calc(n):  
    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: " + str(comp))
```

Result.

The program has been executed and output is verified.

## Output

Enter a number n: 4

The value is : 492

## Program 13

Aim: Sort dictionary in ascending and descending order

## Program

```
import operator
```

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

```
print ('dictionary : ', d)
```

```
s = sorted(d.items(), key=operator.itemgetter(1))
```

```
print ('ascending order : ', s)
```

```
s1 = dict(sorted(d.items(), key=operator.itemgetter(1),  
reverse=True))
```

```
print ('descending order : ', s1)
```

## Result

The program has been ~~verified~~ <sup>executed</sup> and output  
is verified.

## Output

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

in 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 14

Output

Aim: Merge two dictionaries.

Q8. Merge two dict.

### Program

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

Q8. Merge two dict.

Q8. a) Q8(b)

### Result

The program has been executed and output is verified.

Output

None

{'a': 11, 'b': 10, 'c': 11}

## Program 15

Aim : find gcd of two number

Program

Program 15

```
num 1 = int (input ("Enter 1st number :"))
num 2 = int (input ("Enter 2nd number :"))
i=1
while (i<=num1 and i<num2):
    if (num 1 % i == 0 and num 2 % 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 : 80

Enter 2nd number : 20

GCD is 20

## Program 16

factorial

Aim: Program to find the factorial of a number

→ Simple program with

### Program

```
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 output is verified.

Output

Enter a number : 9

The factorial of 9 is 362880

## Program 17

fibonacci

Aim: Generate fibonacci series of n terms.

program

Fibonacci program with

```
nterms = int(input("How many term?"))
n1, n2 = 0, 1
count = 0
if nterms < 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
```

Result

The program has been executed and output is verified.

Output

How many term? 5

fibonacci Sequence:

0  
1  
1  
2  
3

## Program 18

Aim : find the sum of all items in a list

### Program

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

### Result

The program is executed and the output is verified.

1st approach

## Output

2nd approach is used for standard code

How many numbers: 5

Enter number 4

Enter number 2

Enter number 5

Enter number 3

Enter number 5

Sum of elements in given list [4]: 19

## Program 19

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

## Program

Execution with n=3

```
def call():
    n=0
    for x in range(1000,20000):
        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 third % 2 == 0:
                    if fourth % 2 == 0:
                        for i in range(2, number):
                            if i*i == number:
                                print(number)
```

## Result

The program has been executed and output is verified.

Output

>>> call()

4624

6084

6400

((8464

((redman + rokoo) tugen) for  
(redman) appear on  
((redman + rokoo) tugen) for a redman  
(redman) bonggo + tel

((tel) redman + tel comp on the most of redman)

is tugen's soft box business in composition

## Program 20

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

Eg: N=4

1  
2 4  
3 6 9  
4 8 12 16

## Program

```
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=" ")
        print("")
```

## Result

The program has been executed and output was verified.

PI morph

## Output

for a staircase figure with 5 levels is displayed on the  
here rows 5 digit with the other spaces empty  
**>> pyr(5)** displays following a staircase pattern

Enter the number : 5

1  
2 4  
3 6 9  
4 8 12 16  
5 10 15 20 25