

Harithakrishnan

1st year MCA A

Roll No: 40

Amal Jyoti College of
Engineering

Python Lab Record

program NO : 1

AIM : Program to check if else condition.

$a = 12$

$b = 17$

If ($a > b$) :

Print ("the value of a is greater than b")

else :

Print ("the value of b is greater than a")

Result :- The program has been executed and the output was verified.

output

the value of b is greater than a.

Date : 16-01-2021

Program No: 2

AIM: Program to find the square of a number entered by user.

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

```
square = num * num
```

```
print(f"square of {num} is {square}")
```

Result: The program has been executed and the output was verified.

Output

Enter a number q

Square of q is 81

Date : 16 - 01 - 2021

Program No : 3

AIM : Program to find area of a circle using a function.

```
def circleArea(r):
```

```
    PI = 3.14
```

```
    return PI * (r * r)
```

```
n = float(input("Enter radius value"))
```

```
print("area is %.6f", %circleArea(n))
```

Result : The program has been executed and the output was verified.

Output

enter x value : 7

area is 153.860000

Program No : 3

AIM : Program to find the biggest of three numbers entered by the user.

```
num1 = float(input("enter a number:"))
num2 = float(input("enter a number:"))
num3 = float(input("enter a number:"))

if (num1 > num2) and (num1 > num3):
    largest = num1

elif (num2 > num1) and (num2 > num3):
    largest = num2

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

Result : The program has been executed and the output was verified.

Output

enter a number : 3

enter a number : 5

enter a number : 8

the largest number is 8.

Program no: 4

AIM : Count the occurrences of each word in a line of text.

```
def word(sta):
```

```
    counts = dict()
```

```
    words = sta.split()
```

```
    for i in words:
```

```
        if i in counts:
```

```
            counts[i] += 1
```

```
        else:
```

```
            counts[i] = 1
```

```
    return counts
```

```
print(word('have a nice day'))
```

Result : The program has been executed and the output was verified.

Output

```
{'have': 1, 'a': 1, 'nice': 1, 'day': 1}
```

```
((mission is was) (was is) - man  
((mission is was) (is was)) - com  
((mission is was) (was is)) - 89ma  
:(com man) has (man com) 11  
: man = tropical  
:(89ma com) has (com 89ma) 11  
: com = tropical  
((tropical "is com has 89ma 11") - 1000
```

had between and and morning ant [two]
between even things ant

Program No : 5

AIM : Store a list of first names, count the occurrences of 'a' within the list

newline = "say hai anna"

count = 0

for i in newline:

 if i == 'a':

 count = count + 1

print("count of a in say hai anna : " + str(count))

Result : The program has been executed and the output was verified.

Output-

Count of a in say hai anna : 4

Program No : 6

AIM : Get a string from an input string where all occurrences of first character replaced with '\$' except first character.

```
def change_char(stoi):
```

```
    char = stoi[0]
```

```
    stoi = stoi.replace(char, '$')
```

```
    stoi = char + stoi[1:]
```

```
    return stoi
```

```
print(change_char('restart'))
```

Result : The program has been executed and the output was verified.

1920-10-08 00:00

2:00 AM 00:00:00

Output

restart

"moving file 8" - nil

c = false

; switch off i to

; 'o' == 'A'

i + lower = lower

Method "comes of a file not open"; "else"

with bad features and even missing out : Horrible
solution with tuples

Program No: 7

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

```
def change_string(s1):  
    return s1[-1:] + s1[1:-1] + s1[:1]
```

```
print(change_string('abcd'))
```

Result: The program has been executed and the output was verified.

Output

dbca

Program No : 8

AIM : Accept an integer n and compute $n+n+nn$.

$n = \text{int}(\text{input}("enter a number"))$

$tp = \text{str}(n)$

$t1 = tp + tp$

$t2 = tp + tp + tp$

$\text{comp} = n + \text{int}(t1) + \text{int}(t2)$

$\text{print}("value", \text{comp})$

Result : The program has been executed and the output was verified.

Output

Enter a number : 4

Value 399

Program NO : 9

AIM : Merge two dictionaries

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

dict2 = { 'd': 6, 'c': 4 }

```
def merge(dict1, dict2):  
    return dict2.update(dict1))
```

```
print(merge(dict1, dict2))
```

```
print(dict2)
```

Result : The program has been executed and the output was verified.

Output

{'d': 6, 'c': 4, 'a': 10, 'b': 8}

Date : 03-02-2021

Program No : 10

AIM : program to find the factorial of a number

num = int(input("enter a number"))

factorial = 1

If num < 0:

 Print("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 verified.

Output

Enter a number: 7

The factorial of 7 is 5040.

program NO: 11AIM: Generate fibonacci series of N terms

n = int(input("enter the value of 'n': "))

x = int(input("enter the

a = 0

b = 1

sum = 0

count = 1

print("fibonacci series:" end = " ")

while (count <= n):

print(sum, end = " ")

count += 1

a = b

b = sum

sum = a + b

Result: The program has been executed and
the output was verified.

Output

Enter the value of n : 5

0 1 1 2 3

Program No: 12

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

sum = 0

li = [12, 34, 5, 4, 21]

for val in range(0, len(li)):
 sum = sum + li[val]

Print ("sum of all values in a given list",
 sum)

Result: The program has been executed and the output was verified.

Union operation

Output-

Sum of all values in a given list : 76

Program No : 13

AIM : Count the number of characters in a string.

```
def char-freq(str):
    dict = {}
    for n in str:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict
```

```
print(char-freq('pathanamthitta'))
```

Result : The program has been executed and the output was verified.

Output

{'p': 1, 'a': 4, 't': 4, 'b': 2, 'n': 1, 'm': 1
'i': 1}

[18, 4, 2, 4, 81, 91] = 3

:(()) and , a] to make it like this
[18] - 3 + 2 = 18

The output of answer has to make it look
(ans.

it has between each word a space and ; then
brackets will be open

Program No: 14

AIM: Adding 'ing' at the end of a given string.

If it already ends with 'ing' then add 'ly'

```
def add_string(stor):
    length = len(stor)
    if length > 2:
        if stor[-3:] == 'ing':
            stor += 'ly'
        else:
            stor += 'ing'
    return stor
```

Print(add_string('play'))

Print(add_string('Playing'))

Result: The program has been executed and the output was verified.

Output

Playing

Playingly

Program no: 15

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

def longest_word(words_li):

 word_len = []

 for n in words_li:

 word_len.append((len(n), n))

 word_len.sort()

 return word_len[-1][0], word_len[-1][1]

result = longest_word(["rose", "jasmine", "lotus"])

print("longest word : ", result[1])

print("length of the longest word ", result[0])

Result: The program has been executed and the output was verified.

Output : ~~and a to has int has port. int : MIA
has with 'got' does zhas phisics 47 41~~

Longest word : Jasmine

Length of the longest word : 7

: (16+2) print - 16+2 = 73B

(16+2) or = dtprint

: < dtprint >

: 'got' = [: 8 -] 16+2 71

; 52/3

'got' = + 16+2

16+2 auto &

((print)) print - 8+49 ((print))

((print)) print - 8+49 ((print))

Two belongs word and composed int 310232
longer word higher int

Program No: 16

AIM: Generate all factors of a number.

```
n = int(input("enter n"))
```

```
factors = [1]
```

```
for i in range(2, n):
```

```
    if (n % i == 0):
```

```
        factors.append(i)
```

```
        factors.append(n)
```

```
print(factors)
```

Result: The program has been executed and the output was verified.

Output has choices to turn to address : Ma
below - respond to address

enter n 4

[1, 2, 4]

; (1) chooses (brow - respond) - first

[] = ad_brow

; ad_brow on a list

((0, (0) ad)) (choose . ad_brow ~

(3 choose . ad_brow)

[0] [1] ad_brow next

ad_brow ("unusual" "red") (brow - respond -) brows

([1] brows ? : brows - respond ") for 9

brows ? brows - respond to address) brows

values(s) and can compare with : Ma

brows now having with brows

program no : 17

AIM : Lambda functions to find area of a square, rectangle and triangle.

$s = \text{int}(\text{input}("enter the sides : "))$

$x = \lambda a: a * a$

$\text{print}(x(s))$

$t = \text{int}(\text{input}("enter length : "))$

$b = \text{int}(\text{input}("enter breadth : "))$

$y = \lambda l, b: l * b$

$\text{print}(y(t, b))$

$h = \text{int}(\text{input}("enter the height of triangle : "))$

$b_1 = \text{int}(\text{input}("enter the base of triangle : "))$

$z = \lambda h, b_1: (h * b_1) / 2$

$\text{print}(z(h, b_1))$

Result : The program has been executed and the output was verified.

output

enter the sides : 4

16

enter the length : 6

enter breadth : 3

18

enter height of triangle : 7

enter base of triangle : 2

G.O.

(i) baseeqs. exchot

(a) baseeqs. exchot

has between need and mapping part ; thus
bottom was right with

Date : 03-02-2021

Program No : 18

AIM : From a list of integers, create a list removing even numbers.

num = [5, 7, 20, 11, 18, 25, 2]

num = [x for x in num if x%2 != 0]

print(num)

Result : The program has been executed and the output was verified.

for loop example

Output-

[5, 7, 11, 25]

Program No: 19

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

$N = 4$

1

2 4

3 6 9

4 8 12 16

```
def num():
```

```
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(" ")
```

```
num()
```

Result: The program has been executed and the output was verified.

Output

Enter the number : 4

1

2 4

3 6 9

4 8 12 16

Program No : 20

AIM: construct following pattern using nested loop.

```

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
*

```

$n = 5;$

for i in range(n):

 for j in range(i):

 print('*', end='')

 print()

for i in range(n, 0, -1):

 for j in range(i):

 print('*', end='')

 print()

Result : The Program has been executed and the output was verified.

Output

*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*

abc
1
bc
cde
defg

defg

: Output Job

(("Federally ent. valid") -> q1) + 0 = 0

i = i

((1 < 0 , 1) -> q2) or i < 0

i = b

: ((1 < 1 , 1) -> q2) or i < 0

i < i = qm3

((" " has , qm3) -> q4)

((" ") -> q4)

(qm4)

Two statements listed and merging into : Two
bottom two states with