

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

I - НАЯНДАР

Output

Output
Enter first of three numbers : 11
Enter the first number : 2
Enter the second number : 4
Enter the third number : 5 plus = 39

The largest number (840) is 39 workers

bio-fertilizers used and developed

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

9.02.2024

Output signal has 6 outgoing pins

and 6 inputs

Enter an integer Number : 4

Square of 4 is 16

(Address & Value) bus (Value = 16)

(Address & Value) bus (Value = 16)

Address = Report

(Address & Value) bus (Value = 5)

5 address & Report

5 address & Report

The first 5 address reported will be first

and last 5 address reported will be last

base bus and signal and outgoing all

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

E 2. PROGRAM

Output: This is a sample output of the program to find area of circle.

Input the radius of the circle: 4

The area of the circle with radius 4.0 is

50.2654

(Example of Sample I/O sample 2) Note:

for better readability output will
be given in float

PROGRAM : 5

AIM: Python program to find square of n
: given below

list 1 = [14, 20, 13, 8, 6, 2]

for n in list 1:
: print n * n

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

Two boxes need to be mapped with

the first one is two

PROGRAM : 6

AIM: Python Program to find vowels in a string

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

Print ("Given String :\n", string A)

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 the output was verified.

Output

for words having matching vowels:

Hello. How are you? And I said.

The vowels present in the string:

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

Groups of vowels, OI group, AI group

four b's and one m's and
two f's and two t's

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

3 : HA 890099

{'life': 1, 'his': 1, 'a': 1, 'the': 1, 'of': 1, 'memory': 1}

(prob. and word with prob.) list

prob. list (prob. and word) list = col
(elsewhere word list)

prob. list with word elsewhere list (list)

elsewhere list

: fived

prob. list between word and elsewhere list
between word and

PROGRAM - 8

AIM: Python Programs do count a in a list
[a, b, c, d, e, f, g]

a = ['anu', 'akku', 'aashi', 'appu']

sdsds = (''.join(a))

count = 0

for i in sdsds:

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

: (which) found (here)

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 doesnot have equal  
length')
```

Result :

The programs has been executed and output was verified

8 - MAG

Output

list or tuple from oblong and big both list have equal length.

[big, small, both, None]

(None, "D", 1, 6)

o : found

: number n = 3

1 : 1, 6, 2, 3, 4, 5, 6

[4 found, 1 found]

" o : find out no. o for found") true

((None) & & A

both lists are equal and removing off
last item from big

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 lists have equal sum

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

(4 fold) code

Codeless code

Codeless code

Output loops and fold add 10

loops and for loop fail add 10

Output

for loops and and compare

for loops and

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.

01-2-1918

Output:

The results of outputting were

There are common elements

{21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36}

Common elements in the following

(A) ΔD_{down}

(B) ΔD_{down}

(C) ΔD_{down}

(Conc. kept same for all above)

(Conc. kept same for all above)

Following following was not changing

Information about

PROGRAM - 12

AIM: Python Program to replace a character

def change_char (sdn1):

char = sdn1[0]

str1 = sdn1.replace (char, ('\$'))

sdn1 = char + str1 [1:]

Print (change_char ('refresh'))

Result:

The program has been executed and the output was verified.

Output

refresh

[51, 0, 21, 31, 41, 51, 61, 71, 81, 91, 0]

[51, 0, 21, 31, 41, 51, 61, 71, 81, 91, 0]

Initial state

Stage 01

Stage 02

Stage 03

(channels removed and saved)

(channels retained)

it has been reduced from 8 to 5 channels

the remaining channels are

0, 1, 2, 3, 4

PROGRAM : 13

AIM : Python program to 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

Wählen einer Applikation ab entsprechend

$\Rightarrow (3 \times 10^5) \text{ mols}$

[a] 1482

(ϕ , code) unique : 14

Entirely + new

((Cherries)) had a good

PROGRAM - 14

AIM: Python program to merge 2 dictionaries.

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

Returns (dict2) {a: 10, b: 8, c: 2}

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

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

```
Print(merge(dict1, dict2))
```

```
Print(dict2)
```

Result:

The programs has been executed and output was verified.

Output

String of symbols of modified words

None

[None with null string here]

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

: (None) private - open

[0:7]abc + [1-11]abc + [1-12]abc None

(String of 3 private - open)

whose last letter is not part of the string

but first

PROGRAM : 15

AIM: Python Program to ascend and descend dictionary.

Import 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:

$\{(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)\}$ (Gives sorted output)

Dictionary in descending order by value:

$\{3: 4, 4: 3, 1: 2, 2: 1, 0: 0\}$ (Gives sorted output)

$\{(3, 4), (4, 3), (1, 2), (2, 1), (0, 0)\}$ (Gives sorted output)

$\{(3, 4), (4, 3), (1, 2), (2, 1), (0, 0)\}$ (Gives sorted output)

Time taken with each sorting method
+ Python and Java

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

21.1.2023

Output

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

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

remove 21st element from list

[11, 33, 55, 77, 99]

(11, 33, 55, 77, 99)

remove 2nd element from list

(33, 55, 77, 99)

remove 3rd element from list

(55, 77, 99)

remove 4th element from list

(77, 99)

remove 5th element from list

(99)

remove 6th element from list

()

PROGRAM - 17

AIM: Python Program to find gcd of two numbers

```
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 and the output was verified

17: MAR 2024

Output

GCD of 45 and 65 is 5

Java code

for (int i = 1; i <= 45 & i <= 65; i++) {
 if (45 % i == 0 & 65 % i == 0) {
 System.out.println(i);
 }
}

(Ans) 5

Java No 3

Code is 15

(Ans) 15

Ques: Find GCD of 45 and 65
Ans: GCD of 45 and 65 is 5

Java

for (int i = 1; i <= 45 & i <= 65; i++) {
 if (45 % i == 0 & 65 % i == 0) {
 System.out.println(i);
 }
}

(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 leemles 3:5 b mangroves and BB 9:10
estuaries

The factorial of 5's 120

: (3,0,3) pop

: (0,3,3)

Parabola

(d x b d) pop and

27 - 27

27

: ((d,0,3) pop)

((d,0,3) pop, standard, standard, 3:0,0,0) b/w 9:

Q (standard) b/w

with low biomass and and composition
biomes and

biomes and

14 English
AIM: Python program to find Fibonacci sequence

def recurr-fibo(n):

if n <= 1:

return n

else

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

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

if nTerms <= 0:

Print ("Please enter a positive integer")

else:

Print ("Fibonacci Sequence : ")

for i in range(nTerms):

Print (recurr-fibo(i))

Result :

The program has been executed and the output was verified

Output

31 - 1989

How many items? 4

Filousacei Sequence

Concordia D 1963 1 Aug 1963

6

1

1

2

Q. *Alouatta palliata*

PROGRAM - 28

AIM : Python program to performs string function.

```
def add_Edging(sdm):
```

```
    length = len(sdm)
```

```
    if length > 1:
```

```
        if sdm[-3:] == "ing":
```

```
            sdm += 'ly'
```

```
        else:
```

```
            sdm += "ing"
```

```
    return sdm
```

```
Print(add_Edging('do'))
```

```
Print(add_Edging('according'))
```

Result :

The program has been executed and the output was verified.

Output

done
according

Pg - 442/2017

(a) odd-number

Even-number

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

(a) odd-number & (1-a) odd-number

((a) even-number & b)) begin for i=1

((a) odd-number & a)) begin

((a) even-number & b)) begin

((a) even-number & a)) begin

((a) odd-number & b)) begin

PROGRAM - 21

Even digits

AIM : Python Program to generate a list of digit numbers in a given range with all their digits even and the number is Perfect Square.

```
num1 = int(input("Enter the number :"))
num2 = int(input("Enter a number :"))
for i in range(num1, num2+1)
    for j in range(32, 100+1):
        if i == j * j
            string = str(i)
            if int(string[0]) % 2 == 0 and
               int(string[1]) % 2 == 0 and
               int(string[2]) % 2 == 0 and
               int(string[3]) % 2 == 0
            print(i)
```

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

Output

85 - May

Enter a number : 4444

Enter a number : 9999

9624

6084

G A 00

8464

Gibbons = 16 pairs

: "pre" - [C.S.] 1968

$$BV = +1.06$$

PROGRAM - 22

Ques: Python Program to display a pyramid

```
lines = int(input("Enter a number"))
```

```
i = 1
```

```
j = 1
```

```
while i <= lines:
```

```
    j = 1
```

```
    while j <= i:
```

```
        temp = i * j
```

```
        print(temp, end = ' ', flush = True)
```

```
        print(" ", end = ' ', flush = True)
```

```
j = j + 1;
```

```
print(" ");
```

```
i = i + 1;
```

Result :

The program has been executed and the output was verified

Output

Enter a number & it
will print out its square.

1 1 1
2 4 4
3 6 9

(4: 4 * 4 = 16) (5: 5 * 5 = 25)

(6: 6 * 6 = 36) (7: 7 * 7 = 49)

(8: 8 * 8 = 64) (9: 9 * 9 = 81)

(10: 10 * 10 = 100)

1 1 1
2 4 4
3 6 9

4 8 16
5 10 25

6 12 36
7 14 49

8 16 64
9 18 81

10 20 100
11 22 121

12 24 144
13 26 169
14 28 196
15 30 225
16 32 256
17 34 289
18 36 324
19 38 361
20 40 400

PROGRAM : 23

Topic: Python
AIM: Python Program to count the number
of characters in a string.

Code:
`def char_frequency(s):`

`dict = {}`

`for i in s:`

`keys = dict.keys()`

`if i in keys:`

`dict[i] += 1`

`else:`

`dict[i] = 1`

`return dict`

`Print(char_frequency("Hello World"))`

`Result:`

The program has been executed and
the output was verified.

Output
 $\{ 'h': 1, 'e': 1, 'l': 3, 'o': 2, 'd': 2, 'g': 1 \}$

(Output is not displayed)

1 = 1

2 = 2

(Output is not displayed)

1 = 1

1 = 1 attached

1 = 1 attached - open

(Output is not displayed)

(Output is not displayed)

1 = 1

(Output is not displayed)

1 = 1

1 = 1

All lines followed much code except all

PROGRAM - 24

~~QUESTION~~
Ques: Python Program to add 'ing' at the end
of a given string : if already ends with
'ing', then add 'ly'.

def add_string(str1):

length = len(str1)

if length > 1:

if str1[3:] == 'ing':

str1 += 'ly'

else:

str1 += 'ing'

return str1

Result :

The program has been executed
and the output was verified.

E.S. MAGROD

output

walking
walking

Grilled Lemongrass with fish

8.3 = fish

11.060 = fish

Cooked fish = 8.3

Steamed fish

12.463 fish

12.5 fish

12.5 fish

Beef number

(Chicken number 13.000000000000001) fish

for dinner preparation message to
Marilyn new location

PROGRAM - 25

Aim: Python program to accept a list of words and returns length of longest word.

def find(word):

wl = []

for u in word:

wl.append(len(u), u)

wl.sort()

result = wl[-1][0], wl[-1][1]

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

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

Find (["hellow", "instagram", "hi"])

Result:

The program has been executed and
the output was verified

Output

longest word \vdash cursive diagrams

length of the longest word : : 9

(A) $\frac{1}{2}$ mole/liter

(1+62)ad = ad

1960-12-15 - 11.9 fm

PROGRAM : 26

Aim : Python Program to construct nested loop.

```
def start():
    u=5
    for i in range(0):
        for j in range(i):
            print("*", end="")
            print()
    for i in range(0, 0, -1):
        for j in range(0, i):
            print("*", end="")
            print()
    print()
```

Result :

The Program has been executed and
the output was verified

output

which is the output of the model and it
is a categorical variable having class

* *
* *

(Brown) hair

* * *
* * *

ED

* * * * *
* * * * *

brown hair

* * * *
* * *

(Co. Colored) bangs

* *
* *

(Brown)

* right jaw [e] [e] [e] [e] [e] [e] [e]

(E) Jaws with brown hair

lateral temporal with black hair

(Black)

(E) Jaws with black hair

lateral temporal and anterior -

left and right

PROGRAM : 27

Aim: Python Program to generate all factors of a number

Program:

```
def print_factors(x):  
    print("The factors of", x, "are")
```

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

```
    if x % i == 0:
```

```
        print(i)
```

```
print_factors(232)
```

Result:

The program has been executed and
the output was verified.

output

The factors of 232 are

1

2

4

8

29

58

116

232

PROGRAM : 28

AIM: Python program to write function to find area of Square, rectangle and triangle

Print ('Enter the length of a side of Square')

s = int (input ("Enter your value"))

Print ('Enter the length and breadth of rectangle')

l = int (input ("Enter your value"))

b = int (input ("Enter your value"))

Print ('Enter the base and height of triangle')

h = int (input ("Enter your value"))

d = int (input ("Enter your value"))

x = lambda s: s * s

y = lambda l, b: l * b

Output

Final Answer

Enter the length of a side of square.

Enter your value : 2

Enter the length and breadth of rectangle.

Length of rectangle is 5 and breadth is 3.

Area of rectangle is 15.

Perimeter is 16.

(3) Area

(25) Perimeter is 20.

Answer

Area of rectangle and area of square are

both fixed and depend on

$t = 0.5$

$z = \lambda \text{ambda } h, d, t : h * t * d$

Print ("Area of Square", z, es)

Print ("Area of rectangle", y(l, b))

Print ("Area of triangle", z(h, d, t))

Result :

The program has been executed and
the output was verified

PROGRAM : 29

Aim: Python program to create rectangle class with attributes length and breadth and method to find area and perimeter. Compare two rectangle objects by their area.

class rectangle

def __init__(self, l, b)

self.length = l

self.breadth = b

def area(self):

Returns self.length * self.breadth

def Perimeter(self):

Returns 2 * (self.length + self.breadth)

def cmp(self, obj)

if self.area() > obj.area():

Print ("Rectangle with length = ", self.length
and "b breadth = ", self.breadth, ' has the
greater area')

else self.area() > obj.area() then print
("The rectangle has the greater area")

Print ("Rectangle width length = ", self.length,
and breadth, ' has the greater one')
else print ("The rectangle has the greater area")

Print ("They have equal area!")

γ_1 = rectangle (9, 3)

γ_2 = rectangle (3, 4)

$\gamma_1 \text{ cmp } (\gamma_2)$

Result :

The program has been executed and
the output was verified.

Output

Length : 4.6 is obtained
Rectangle with length = 9 and breadth = 3

has greater area.

(Circles are repeated for next 3 boxes)

for finding out the area of rectangle with
length and width.

length and width with

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

PROGRAM : 30

AIM: Python Program to create a bank account with members account number, name, type of account and balance . create constructor and method to deposit at the bank and withdraw an amount from the bank

class BankAccount :

def __init__(self, a, n, t, b) :

self.acno = a

self.name = n

self.type = t

self.bal = b

def deposit(self, a) :

self.bal += a

Print ('Rs', a, 'deposited' 'current balance
is: Rs', self.bal)

def withdraw(self, a) :

if self.bal >= a:

 self.bal -= a

Point ('Rs', a, 'Withdraw' current balance is
R'', self.bal)

Point ('Insufficient balance to make this
transaction!')
print('Insufficient balance to make this
transaction!')

a = int(input('Enter account number'))

b = input('Enter name of the account holder')

t = input('Enter account type')

b = float(input('Enter your balance'))

acc = Bank Account(a, n, t, b)

acc.deposit(float(input('Enter amount
to deposit :')))

acc.withdraw(float(input('Enter amount
to withdraw :')))

Result :

The program has been executed and
the output was verified.

Output

Enter name of account holder : John

Enter account number : 725492342

Enter account type : Savings

Enter your balance : 1000000

Enter amount to deposit : 3000000

Rs. 3000000.0 deposit : current balance is

Rs 4000000

Enter amount withdraw 5000.

Rs. 5000 withdraw ! current balance is

395000.0

PROGRAM : 31

Ans: Python program to create a class Rectangle with private attributes length and width. overload ‘`c`’ operator to compare the area of 2 rectangle.

Class Rectangle :

```
def __init__(self, l, w):
```

```
    self.length = l
```

```
    self.width = w
```

```
    self.area = self.width * self.length
```

```
def __lt__(self, other):
```

```
if self.area < other.area:
```

```
Print('Rectangle with length = ', self.width)
```

```
= ', self.length, ' has the lower area', )
```

```
elif other.area < self.area:
```

```
Print('Rectangle with length = ', other - -
```



```
length, and width = ', other -- width, ) =
```

other -- width , ' has the larger area ! ')

else :

Print (' They have equal area ! ')

$l = \text{float}(\text{input}(\text{'Enter length of 1st rectangle'}))$

$w = \text{float}(\text{input}(\text{'Enter width of 1st rectangle'}))$

$R_1 = \text{Rectangle}(l, w)$

$l = \text{float}(\text{input}(\text{'Enter length of 2nd rectangle'}))$

$w = \text{float}(\text{input}(\text{'Enter width of 2nd rectangle'}))$

$R_2 = \text{Rectangle}(l, w)$

$R_1 < R_2$

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

Output

Enter length of 1st rectangle : 2

Enter width of 1st rectangle : 3

Enter the length of 2nd rectangle : 5

Enter width of 2nd rectangle : 3

Rectangle with length = 2.0 cm
and width = 3.0 cm has the area 6.0 cm².

Enter length of 3rd rectangle : 6

Enter width of 3rd rectangle : 4

Rectangle with length = 6.0 cm
and width = 4.0 cm has the area 24.0 cm².

(Addition of rectangles)

lengths which required for 3 rectangles

(6.0 + 4.0 + 2.0) cm

length of 1st rectangle

(6.0 + 4.0 + 2.0) cm

length of 2nd rectangle

PROGRAM : 32

Aim: Python Programs do create a class Time with private attributes hours, minutes and seconds, overload '+' operator do find sum of 2 time.

class Time :

def __init__(self, hh=0, mm=0, ss=0):

self.hour = hh

self.minute = mm

self.second = ss

def __add__(self, other):

second = int((self.__second + other.__second) % 60)

minute = int(((self.__minutes + other.__minutes) * 60 + (self.__second + other.__second)) / 60)

hours = int((self.__hour + other.__hour) * 24 + (self.__minutes + other.__minutes) / 60)

Print('Time [hh,mm,ss]', 'hour', ':', minute, ':', second]

$T_1 = \text{Time}(12, 25, 45)$

$T_2 = \text{Time}(16, 45, 56)$

$T_1 + T_2$

Result :

The program has been executed and
the output was verified

Output

— Line 1 (cont. and cont'd.)
Time [hh:mm:ss] S: 11:48
Approximate notes (approx. 650)
[approx. 650] [approx. 650] Logarithmic

(cont'd.)

Comments for approx. notes (approx.) to

the following numbers (approx.) by

Approximate

Cont'd. Approximate

Comments for approx. notes (approx.) to

the following numbers (approx.) by

PROGRAM - 33

Aim: Python program to create a class Publisher (name) Derive class Book from Publisher with attributes title and author. Derive class python from book with attributes Price and no of Pages write a program to that display information about a python book like here class constructor invocation and method overriding.

class publisher :

```
def __init__(self, name):
```

```
    self.name = name
```

```
def show(self):
```

```
    pass
```

class Book (publisher) :

```
def __init__(self, title, author, name):
```

```
    self.title = title
```

Self . author = author

Publisher . --init__(self , name)

def show(self):

Pass.

class Python(Book):

def __init__(self , p , no , title , author , name):

Self . price = p

Self . no . of - Pages = no

Book . --init__(self , title , author , name)

def show(self):

Print('Book title : ', self . title)

Print('Author : ', self . author)

Print('Publisher : ', self . publisher)

Print('Price . RS : ', self . price)

Print('No of pages : ', self . no of Pages)

P = Python(999.99 . 400 , 'programming with
Python' , 'George Eliot' , 'ABC Book')
P . show()

Result :

The program has been executed and
the output was verified

Output

(74, 67, 51) and

Book title : programming with python

Author : George Elst

Publisher : ABC Book.

Price : RS. 565.9

for beginners need code examples

No of page : 250

PROGRAM : 34

AIM: Python program to read a file line by line and store it into a list

```
def file_read(fname):
    f = open(fname)
    content_list = []
    for line in f:
        content_list.append(line)
    f.close()
    return content_list
```

```
c = file_read("demo.txt")
```

```
print(c)
```

```
#Print len(c)
```

```
file_read ("demo.txt")
```

Result

The program has been executed and the output was verified

Output

[' There was a banyan tree in which
he had a variety of birds and he got
it from a Squirrel (described it). Two
of them were very interesting. They
were very good singers.

Common myna, black faced lanius, etc.
Cuckoo (Cuculus) was also seen.

Two other species were seen.

PROGRAM : 35

Aim : Python program do copy odd lines of one file to other.

```
a = open('demo.txt', 'r')
```

```
b = open('f.txt', 'w')
```

```
c = a.readlines()
```

```
for i in range(0, len(c)):
```

```
    if (i % 2 != 0):
```

```
        b.write(c[i])
```

```
else:
```

```
    pass
```

```
b.close
```

```
b = open('f.txt', 'r')
```

```
d = b.read()
```

```
print(d)
```

```
a.close()
```

```
b.close()
```

Result :

The program has been executed and the output was verified

Output

of a form of mapping with β .
 They are distinguished from dump trucks by configuration: a dump truck has an open-closed vehicles with the leadership in front of the driver with a tail following.

and back

combinations

(C00) standard

(C00) articulated

(C00, articulated) back, at

front

with back, front and middle supports

laid down under the body

PROGRAM : 36

AIM: Python programs do read each row from a given CSV file and print a list of strings.

```
import csv  
with open('dep.csv', newline = '') as csvfile:  
    d = csv.reader(csvfile, delimiter = ',',  
                   quotechar = "'")  
    for r in d:  
        print(r, 'String')
```

Result :

The program has been executed and the output was verified

Output

[[11, 12, 13], "[33, 25, 56]", [55, 20, 31],
 "[13, 25, 40]", "[30, 20, 50]", [25, 30, 10],
 "[40, 20, 50]", "[50, 160, 10]", [30, 40, 10]]

: (0,0) at (0,0) of axis 001
: (0,1) at (0,1)

(100,0) at (0,0)

Ques. What is Python?

Object Oriented

Scripting

Open Source

Free

Platform Independent

Large standard library

PROGRAM : 37

AIM : Python programs perform the sum of given numbers.

```
numbers = [1, 2, 3, 4, 5, 2, 5]
```

```
sums = sum(numbers)
```

```
print(sums)
```

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

DEPARTMENT

Output

the output for aboriginal people in
the full 22 hours found 20 different

types of artefacts

of which 15 were

newly described, 1020 got 'Design' status

and 1000 had 'Stylized' characters

(Y-shaped lines)

bow and arrow

(Circles) 13000

lines

the last one could not measure up

but from other angles it's

PROGRAM : 38

AIM : Python programs do print out all colors from color-list1 not contained in color-list2.

Color 1 - Set (Input ('Enter colors separated by commas : ')). split (',')

Color 2 - Set (Input ('Enter colors separated by commas : ')). split (',')

Print ('colors in color-list1 not contained in color-list2 are : list (color1)
difference (color2))

Result :

The program has been executed and the output was verified

Output

Color colors Separated by commas: red, yellow
brown.

Color colors Separated by commas: black,
color in color list 1 and color in:
color list 2 are: [brown, red, yellow]

Color colors Separated by commas: black

brown, red, yellow

PROGRAM NO: 39

Ques: Python programs do generate positive list of numbers from a given list of integers.

```
list1 = [1, -1, 2, -5, 9, -2, -54, 87, -33, 76,  
24, -67]
```

Method: using for loop

```
pos = list()
```

```
for i in list1  
if i > 0
```

```
pos.append(i)
```

```
Print("original list ", list1)
```

```
Print("positive integer list : ", pos)
```

Result:

The program has been executed and the output was worked

Output

original list: $[1, -1, 2, -5, 9, -2, e^{-5}, 4, 8, \frac{1}{7}, -\frac{3}{2}, 76, 24, -67]$

Positive integer list : [1, 2, 9, 8, 7, 24]

On 20th Aug. 1967 I had noted
the following. On 20th

PROGRAM: 40

Ans: Python programs do create a list of colors from comma-separated color names entered by user display first and last color.

Colors = Input ('Enter colors separated by commas : ') . split (',')

Print ('first color : ', colors [0])

Print ('last color : ', colors [len (colors) - 1])

Result :

The program has been executed and the output was verified.

Output

Color color Separated by commas : red, blue,
green, yellow

first color : red

last color : yellow

[red, blue,

green, yellow]

list of colors

list of colors