Python programming-CSA0814

DAY 9

(21 aug 24)

**1.PRINTING NUMBERS WHICH IS DIVIDED BY 3&5 UPTO 200**

For i in range(1,200):

If i%3==0 or i%5==0:

print(i)

**OUTPUT:**

3

5

6

9

10

12

15

18

20

21

24

25

27

30

33

35

36

39

40

42

45

48

50

51

54

55

57

60

63

65

66

69

70

72

75

78

80

81

84

85

87

90

93

95

96

99

100

102

105

108

110

111

114

115

117

120

123

125

126

129

130

132

135

138

140

141

144

145

147

150

153

155

156

159

160

162

165

168

170

171

174

175

177

180

183

185

186

189

190

192

195

198

**2.SUM OF ROWS,COLUMNS,DIAGONAL ELEMENTS IN MATRIX**

import numpy as n

a=[[1,2,3],

[4,5,6],

[7,8,9]]

x=n.array(a)

print("row sum",n.sum(x,axis=1))

print("columnsum",n.sum(x,axis=0))

print("diagonalsum",n.sum(n.diag(x)))

OUTPUT:

row sum [ 6 15 24]

columnsum [12 15 18]

diagonalsum 15

**3.SUM OF DIGITS**

num=121

sod=0

while num>0:

r=num%10

sod+=r

num//=10

print(sod)

**OUTPUT:**

4

**4.ADD THE MAX NUMBER IN GIVEN TRIANGLE PATTERN**

t = [

[3],

[4, 6],

[2, 7, 6],

[8, 5, 9, 6]]

n = [max(sublist) for sublist in t]

s = 0

for j in n:

s += j

print(s)

**OUTPUT:**

25

**5.ADDING THE STRING SEQUENCE**

x={1:['a','b'],2:['c','d']}

y=[]

for i in x[1]:

for j in x[2]:

y.append(i+j)

print(' '.join(y))

**OUTPUT:**

ac ad bc bd

**6.FIND MAXIMUM PALINDROME**

max\_palindrome = 0

for i in range(9999, 999, -1):

for j in range(i, 999, -1):

product = i \* j

if product <= max\_palindrome:

break

if str(product) == str(product)[::-1]:

max\_palindrome = product

print(max\_palindrome)

**OUTPUT:**

99000099

**7.DIFFERENCE B/W SUM OF SQ. AND SQ. OF SUM UPTO 200**

n=200

sumsq=sum(i\*\*2 for i in range(1,n+1))

sqsum=sum(range(1,n+1))\*\*2

diff=sqsum-sumsq

print(diff)

**OUTPUT:**

401323300

**8.ADDICTIVE SEQUENCE**

def addic(seq):

return all(seq[i]==seq[i-1]+seq[i-2] for i in range(2,len(seq)))

sequence=[6,6,12,18,30]

print(addic(sequence))

**OUTPUT:**

True

**9.GEOMETRIC SEQUENCE OR NOT**

def geo(seq):

if len(seq)<2:

return true

ratio=seq[1]/seq[0]

return all(seq[i]/seq[i-1]==ratio for i in range(2,len(seq)))

sequence=[2,6,18,54]

print(geo(sequence))

**OUTPUT:**

True

**10.JUSTIFY THE SENTENCE**

def justify(sen, spaces):

spacestr = ' ' \* spaces

words = sen.split()

justified\_sen = spacestr.join(words)

return justified\_sen

sentence = input("Enter the sentence: ")

space = int(input("Enter the number of spaces for each word: "))

result = justify(sentence, space)

print("\nJustified sentence:\n")

print(result)

**OUTPUT:**

Enter the sentence: dwcx yus yudyuq f ydf f cyuqf xu c uwqfd yu xvyu acc qd x qcyu cy qcy cyu c gyg yu gy c yuc gu ggd xd c xv dyu fYU XUA S UYfyuf ua gcyu y xydqva wdq yuwdqg xy g

Enter the number of spaces for each word: 5

Justified sentence:

dwcx yus yudyuq f ydf f cyuqf xu c uwqfd yu xvyu acc qd x qcyu cy qcy cyu c gyg yu gy c yuc gu ggd xd c xv dyu fYU XUA S UYfyuf ua gcyu y xydqva wdq yuwdqg xy g