27-04-2021

python programs

Flat list:

```
l = [[1,2], [2,3], 3]
result = []

def fun(l):
    for i in l:
        if type(i) == list:
            fun(i)
        else:
            result.append(i)
    return result

x = fun(l)
print(x)
```

Output:

```
[1, 2, 2, 3, 3]
```

Reverse a given number:

```
num = 123456789
rev =0

while num>0:
        rev = rev * 10 + num % 10
        num = num//10

print(rev)
```

Output

```
987654321
```

Swap comma with dot and dot with comma

```
def Replace(str1):
  str1 = str1.replace(', ', 'hi')
  print(str1)
  str1 = str1.replace('.', ', ')
  str1 = str1.replace('hi', '.')
  return str1

string = "14, 625, 498.002"
print(Replace(string))
```

Output:

14.625.498, 002

First five odd numbers

```
l = []
for i in range(0,100):
        if i%2!=0:
                l.append(i)
                if len(l)==5:
                        break
print(l)
```

Output
[1, 3, 5, 7, 9]

Reverse a string using recursive function:

```
def reverse(s):
  if len(s) == 0:
    return s
  else:
    return reverse(s[1:])+s[0]

s = 'abc'
res = reverse(s)
print(res)
```

Output:

cba

print numbers in right triangle in python

```
n = int(input('enter the number of rows'))
num = 1
for row in range(1,n+1):
  for col in range(1, row+1):
     print(num, end='')
     num = num+1
  print()
```

Output:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

Reverse a list without using [::-1]

```
lst = [10, 11, 12, 13]
last_idx = len(lst)-1
print(last_idx)
for i in range(len(lst)):
  lst.insert(i, lst.pop(last_idx))
print(lst)
```

Output:

```
[13, 12, 11, 10]
```

Remove extra spaces in a string

```
import re
Input= "abcd       pq rs     qwe   "
new_str = re.compile(r"\s+")
my_str = new_str.sub(" ",Input).strip()
print(my_str)
```

Output

```
abcd pq rs qwe
```

Leap year:

```
year = int(input("enter a year: "))
if (year%4) == 0:
  if(year%100)==0:
    if(year%400)==0:
      print('{} is a leap year'.format(year))
    else:
      print('{} is not a leap year'.format(year))
  else:
    print("{} is leap year". format(year))
else:
  print("{} is not a leap year".format(year))
```

Output:

```
enter a year: 1992
1992 is leap year
```

Prime number:

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

if n > 1:
  for i in range(2,n):
    if(n%i) == 0:
      print(n, "is not a prime number")
      break

  print(n, 'is a prime number')
else:
  print(n, 'is not a prime number')
```

Output:

```
enter a number:11
11 is a prime number
```

Wordcount program

```
def myfunction(num):
  wordtonum={'eight':8, 'two':2}
  numword = input('enter number in word')
  m = num*wordtonum[numword]
  return m
```

Output:

```
enter a number8
enter number in wordeight
64
```

Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints: Consider use range(#begin, #end) method

```
l =[]
for i in range(2000, 3201):
  if(i%7==0) and (i%5!=0):
     l.append(str(i))

print(','.join(l))
```

Output:

2002,2009,2016,2023,2037,2044,2051,2058,2072,2079,2086,2093,2107,2114,2121,2128,2142,
2149,2156,2163,2177,2184,2191,2198,2212,2219,2226,2233,2247,2254,2261,2268,2282,2289,
2296,2303,2317,2324,2331,2338,2352,2359,2366,2373,2387,2394,2401,2408,2422,2429,2436,
2443,2457,2464,2471,2478,2492,2499,2506,2513,2527,2534,2541,2548,2562,2569,2576,2583,
2597,2604,2611,2618,2632,2639,2646,2653,2667,2674,2681,2688,2702,2709,2716,2723,2737,
2744,2751,2758,2772,2779,2786,2793,2807,2814,2821,2828,2842,2849,2856,2863,2877,2884,
2891,2898,2912,2919,2926,2933,2947,2954,2961,2968,2982,2989,2996,3003,3017,3024,3031,
3038,3052,3059,3066,3073,3087,3094,3101,3108,3122,3129,3136,3143,3157,3164,3171,3178,
3192,3199

Write a program which can compute the factorial of a given number. The results should be printed in a comma-separated sequence on a single line. Suppose the following input is supplied to the program: 8
Then, the output should be: 40320

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

```
def fact(x):
  if x==0:
     return 1
  return x*fact(x-1)

x=int(input('enter a number'))
print(fact(x))
```

Output:

```
enter a number5
120
```

With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary. Suppose the following input is supplied to the program: 8 Then, the output should be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. Consider use dict()

```
n = int(input('enter a number'))
d=dict()
for i in range(1, n+1):
   d[i]= i*i
print(d)
```

Output:

```
enter a number5
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program: 34,67,55,33,12,98 Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34', '67', '55', '33', '12', '98')

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. tuple() method can convert list to tuple

```
values = input('enter a numbers with comma seperated')
l=values.split(',')
t=tuple(l)
print(l,t)
```

enter a numbers with comma separated 12,23,34,34
['12', '23', '34', '34'] ('12', '23', '34', '34')

Prime number and Fibonacci number :

```
num =11

for i in range(2, num//2+1):
   if num%i==0:
      print(num, 'is not a prime number')
      break
else:
   print(num, 'is a prime number')
```

Output:

11 is a prime number

Fibonacci

```
a=0
b=1
for i in range(1,10):
  print(a)
  a,b=b,a+b
```

Output:

```
0
1
1
2
3
5
8
13
21
```

Remove vowels:

```
def remove_vowels(string):
  vowels = ('a', 'e', 'i', 'o','u')
  for each in string.lower():
     if each in vowels:
        string = string.replace(each, '')
  return string

string = ' this is a string'
print(remove_vowels(string))
```

Output

```
 ths s  strng
```

Write a function
def solution(N)

That, given an integer N(1<=N<=100), returns an array containing N unique integers that sum up to 0. The function can return any such array.

For example, given N=4, the function could return[1,0,-3,2] or [-2,1,-4,5]. The answer[1,-1,1,3] would be incorrect(because value 1 occurs twice). For N=3 one of the possible answers is [-1, 0,1](but there are many more correct answers).

```
import random as r
N = int(input('enter a number between 1 and 100:'))
flag=0
while flag==0:
  numbers = []
  for x in range(N):
     flag1=0
     while flag1==0:
        random_num =r.randint(-1000, 1000)
        if random_num not in numbers:
           flag1=1
           numbers.append(random_num)
  if sum(numbers)==0:
     flag=1000
     print(numbers)
```

Output

```
enter a number between 1 and 100:10
[-4, 592, -444, -380, 71, 153, 387, -605, 715, -485]
```

Days of the week are represented as three-letter strings(‘mon’, ‘tue’, ‘wed’, ‘thu’, ‘fri’, ‘sat’, ‘sun’)

Write a function solution that given a string S representing the day of the week and an integer K (between 0 and 500, inclusive), returns the day of the week that is K days later. For example, given S=”wed” and K =2, the function should return “fri”

Given s = “sat” and k=23 the function should return “mon’

```
days = ['mon','tue','wed','thu','fri','sat','sun']
s=input('enter a day')
k=int(input('enter the number of days'))
ind=days.index(s)
print(ind)
f=(k+ind)%7
print(f)
print(days[f])
```

Output

```
enter a daywed
enter the number of days2
2
4
fri
```

Define a class which has at least two methods: getString: to get a string from console input printString: to print the string in upper case. Also please include simple test function to test the class methods.

Hints: Use __init__ method to construct some parameters

```
  def __init__(self):
      self.s = ''
  def getString(self):
      self.s = input('enter a string in lower case')
  def printString(self):
      print(self.s.upper())

strObj = InputOutString()
strObj.getString()
strObj.printString()
```

Output

```
enter a string in lower case test
 TEST
```