Q1: 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.

In [3]:
```
a=[]
for i in range(2000,3201):
    if i%7==0 and i%5!=0:
      a.append(str(i))
print(','.join(a))
```

```
2002,2009,2016,2023,2037,2044,2051,2058,2072,2079,2086,2093,2107,2114,2121,2128,21
42,2149,2156,2163,2177,2184,2191,2198,2212,2219,2226,2233,2247,2254,2261,2268,228
2,2289,2296,2303,2317,2324,2331,2338,2352,2359,2366,2373,2387,2394,2401,2408,2422,
2429,2436,2443,2457,2464,2471,2478,2492,2499,2506,2513,2527,2534,2541,2548,2562,25
69,2576,2583,2597,2604,2611,2618,2632,2639,2646,2653,2667,2674,2681,2688,2702,270
9,2716,2723,2737,2744,2751,2758,2772,2779,2786,2793,2807,2814,2821,2828,2842,2849,
2856,2863,2877,2884,2891,2898,2912,2919,2926,2933,2947,2954,2961,2968,2982,2989,29
96,3003,3017,3024,3031,3038,3052,3059,3066,3073,3087,3094,3101,3108,3122,3129,313
6,3143,3157,3164,3171,3178,3192,3199
```

Q2: Write a program which can compute the factorial of a given numbers. The results should be printed in a comma-separated sequence on a single line. Suppose the following input is supplied to the program:

In [2]:
```
a=int(input())
b=1
while a-1>0:
    b=b*a
    a=a-1

print(b)
```

```
40320
```

Q3: 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}

In [12]:
```
a = int(input())
b={}

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

```
    b[i]=i*i

else:
    pass

print(b)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

Q4: 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')

In [1]:
```
a=input()
b=a.split()
c=tuple(b)
print(b)
print(c)
```

```
['34,67,55,33,12,98']
('34,67,55,33,12,98',)
```

Q5: 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.

In [25]:
```
class A(object):

    def __init__(self):
            self.str = ""
    def uppercase(self):
            self.str = input()
            print(self.str.upper())

function = A()
function.__init__()
function.uppercase()
```

```
HOU
```

Q6: Write a program that calculates and prints the value according to the given formula: Q = Square root of [(2 C D)/H] Following are the fixed values of C and H: C is 50. H is 30. D is the variable whose values should be input to your program in a

comma-separated sequence. Example Let us assume the following comma separated input sequence is given to the program: 100,150,180 The output of the program should be: 18,22,24

In [58]:
```
import math
C = 50
H = 30
D = input().split(',')
A=[]
for num in D:
 E = (2*C*int(num)) / H
 Formula_Q = math.sqrt(E)
 A.append(int(Formula_Q))
print(A)
```

```
[18, 22, 24]
```

Q7: Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i*j. Note: i=0,1.., X-1; j=0,1,¡Y-1. Example Suppose the following inputs are given to the program: 3,5 Then, the output of the program should be: [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

In [ ]:
```
import numpy
```

Q8: Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. Suppose the following input is supplied to the program: without,hello,bag,world Then, the output should be: bag,hello,without,world

In [61]:
```
a = input("Please, input words")
b_list = a.split(',')
b_list.sort()
print(',' .join(b_list))
```

```
bag,hello,without,world
```

Q9: Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized. Suppose the following input is supplied to the program: Hello world Practice makes perfect Then, the output

should be: HELLO WORLD PRACTICE MAKES PERFECT

In [112…
```
lines = []
b = str(input("Please, input words"))
if b:
    lines.append(b)

for a in lines:
    print(a.upper())
```

```
HELLO WORLD PRACTICE MAKES PERFECT
```

Q10: Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the following input is supplied to the program: hello world and practice makes perfect and hello world again Then, the output should be: again and hello makes perfect practice world

In [119…
```
a = input().split(' ')
b=[]
for c in a:
    if c not in b:
        b.append(c)
    else:
        continue
b.sort()
print(" " .join(b))
```

```
again and hello makes perfect practice world
```

Q11: Define a class, which have a class parameter and have a same instance parameter.

In [48]:
```
class CAR(object):
    brand = "CAR"
    def __init__(self, brand):
         self.brand = brand
    def CAR(self, power):
         self.power = power
mustang = CAR("mustang")
mustang.power = 400
print("Car is %s and power is %s" % (mustang.brand, mustang.power))
```

```
Car is mustang and power is 400
```

Q12: Define a function which can generate a dictionary where the keys are numbers between 1

and 20 (both included) and the values are square of keys. The function should just print the values only.

In [78]:
```
def printnumber():
    square = {}
    for i in range(1,21):
        square[i]=i*i

    print(square.values())

printnumber()
```

```
dict_values([1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 2
89, 324, 361, 400])
```

Q13: Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys. The function should just print the keys only.

In [79]:
```
def printnumber():
    square = {}
    for i in range(1,21):
        square[i]=i*i

    print(square.keys())

printnumber()
```

```
dict_keys([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20])
```

Q14: Define a function which can generate a list where the values are square of numbers between 1 and 20 (both included). Then the function needs to print the first 5 elements in the list.

In [83]:
```
def printlist():
    a = []
    for i in range(1,21):
        a.append(i*i)
    print(a[:4])

printlist()
```

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

Q15: Write a program to generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

In [91]:
```
a = (1,2,3,4,5,6,7,8,9,10)
b = []
for i in a:
```

```
    if int(i)%2 == 0:
        b.append(i)
c=tuple(b)
print(c)
```

```
(2, 4, 6, 8, 10)
```

Q16: Define a class named American and its subclass NewYorker

In [93]:

```
class American(object):
      pass
class NewYorker(American):
      pass

oneAmerican = American()
oneNewYorker = NewYorker()
print(oneAmerican)
print(oneNewYorker)
```

```
<__main__.American object at 0x7f2858847f10>
<__main__.NewYorker object at 0x7f2858847b20>
```

Q17: Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.

In [8]:

```
class Rectangle(object):

    def __init__(self, length, width):
        self.length = length
        self.width = width

    def dim(self):
         return self.length*self.width


aRectangle = Rectangle(4,6)
print(aRectangle.dim())
```

```
24
```

Q18: Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default.

In [9]:

```
class Shape(object):

    def _init__(self, area):
        self.area = 0

class Square(Shape):

    def __init__(self, length):
        self.length = length
```

```
    def a(self):

        return self.length*self.length


aSquare = Square(5)
print (aSquare.a())
```

```
25
```

Q19: Write a function to compute 5/0 and use try/except to catch the exceptions.

In [16]:
```
def calculation():
    return 5/0

    try:
        calculation()

    except TypeErr():
          print("Type is Error")
```

Q20: With a given list [12,24,35,24,88,120,155,88,120,155], write a program to print this list after removing all duplicate values with original order reserved.

In [155…
```
a = [12,24,35,24,88,120,155,88,120,155]
b = []
for i in a:
    if i not in b:
         b.append(i)
print(b)
```

```
[12, 24, 35, 88, 120, 155]
```

Q21: Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence. Example: 0100,0011,1010,1001 Then the output should be: 1010 Notes: Assume the data is input by console.

In [130…
```
a = input().split(',')
b=[]
for c in a:
    if int(c) % 5==0:
        b.append(c)
    else:
        continue
print(',' .join(b))
```

```
0100,1010
```

Q22: Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. The numbers obtained should be printed in a comma-separated sequence on a single line.

In [18]:
```
a = []
for b in range(1000,3001):
    i = str(b)
    if (int(i[0])%2==0) and (int(i[1])%2==0) and (int(i[2])%2==0) and (int(i[3])%2=
        a.append(i)
    else:
        continue
print(','.join(a))
```

```
2000,2002,2004,2006,2008,2020,2022,2024,2026,2028,2040,2042,2044,2046,2048,2060,20
62,2064,2066,2068,2080,2082,2084,2086,2088,2200,2202,2204,2206,2208,2220,2222,222
4,2226,2228,2240,2242,2244,2246,2248,2260,2262,2264,2266,2268,2280,2282,2284,2286,
2288,2400,2402,2404,2406,2408,2420,2422,2424,2426,2428,2440,2442,2444,2446,2448,24
60,2462,2464,2466,2468,2480,2482,2484,2486,2488,2600,2602,2604,2606,2608,2620,262
2,2624,2626,2628,2640,2642,2644,2646,2648,2660,2662,2664,2666,2668,2680,2682,2684,
2686,2688,2800,2802,2804,2806,2808,2820,2822,2824,2826,2828,2840,2842,2844,2846,28
48,2860,2862,2864,2866,2868,2880,2882,2884,2886,2888
```

Q23: Write a program that accepts a sentence and calculate the number of letters and digits. Suppose the following input is supplied to the program: hello world! 123 Then, the output should be: LETTERS 10 DIGITS 3

In [11]:
```
A = ['0','1','2','3','4','5','6','7','8','9']
B = ['a','b','c','d','e','f','g','h','j','k','l','m','n','o','p','q','r','s','t','
c = input()
DIGIT = 0
LETTER =0
for i in c:
    if i in A:
        DIGIT += 1
    if i in B:
        LETTER += 1
    else:
        pass

print("LETTERS", LETTER)
print("DIGIT", DIGIT)
```

```
LETTERS 10
DIGIT 3
```

Q24: Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters. Suppose the following input is

supplied to the program: Hello world! Then, the output should be: UPPER CASE 1 LOWER CASE 9

In [14]:
```
a = input()
UPPER_CASE = 0
LOWER_CASE = 0
for i in a:
    if  i.isupper():
        UPPER_CASE +=1
    if  i.islower():
        LOWER_CASE +=1
    else:
        pass
print("UPPER CASE", UPPER_CASE)
print("LOWER CASE", LOWER_CASE)
```

```
UPPER CASE 1
LOWER CASE 9
```

Q25: Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a. Suppose the following input is supplied to the program: 9 Then, the output should be: 11106

In [16]:
```
b = int(input("a="))
a = b
aa = 10*b+a
aaa = 100*b + aa
aaaa = 1000*b + aaa
c = a+aa+aaa+aaaa
print(c)
```

```
11106
```

Q26: Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers. Suppose the following input is supplied to the program: 1,2,3,4,5,6,7,8,9 Then, the output should be: 1,3,5,7,9

In [25]:
```
a = input().split(',')
b = []
for i in a:
    if int(i) %2!=0:
        b.append(i)
    else:
        continue
print(",".join(b))
```

```
1,3,5,7,9
```