ASSIGNMENT 1,

Name: ENG KHUN

ID: M080101

Subject: Programs for Data Science

Question 1

Level 1

Question:
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

In [1]:

```
for i in range (2000, 3201):
    if i%7 == 0 and i%5 != 0:
        print(i, end = ', ')
```

```
2002, 2009, 2016, 2023, 2037, 2044, 2051, 2058, 2072, 2079, 2086, 2093, 210
7, 2114, 2121, 2128, 2142, 2149, 2156, 2163, 2177, 2184, 2191, 2198, 2212, 2
219, 2226, 2233, 2247, 2254, 2261, 2268, 2282, 2289, 2296, 2303, 2317, 2324,
2331, 2338, 2352, 2359, 2366, 2373, 2387, 2394, 2401, 2408, 2422, 2429, 243
6, 2443, 2457, 2464, 2471, 2478, 2492, 2499, 2506, 2513, 2527, 2534, 2541, 2
548, 2562, 2569, 2576, 2583, 2597, 2604, 2611, 2618, 2632, 2639, 2646, 2653,
2667, 2674, 2681, 2688, 2702, 2709, 2716, 2723, 2737, 2744, 2751, 2758, 277
2, 2779, 2786, 2793, 2807, 2814, 2821, 2828, 2842, 2849, 2856, 2863, 2877, 2
884, 2891, 2898, 2912, 2919, 2926, 2933, 2947, 2954, 2961, 2968, 2982, 2989,
2996, 3003, 3017, 3024, 3031, 3038, 3052, 3059, 3066, 3073, 3087, 3094, 310
1, 3108, 3122, 3129, 3136, 3143, 3157, 3164, 3171, 3178, 3192, 3199,
```

Question 2

Level 1

Question: 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: 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.

In [2]:

```
numb = input()
numb = int(numb)
x = 1
for i in range(1,numb+1):
    x = x*i
print(x)
```

```
8
40320
```

Question 3

Level 1

Question: 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()

In [4]:

```
numb = int(input())
ans = {}
for i in range(1,numb+1):
    ans[i] = i*i
print(ans)
```

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

Question 4

Level 1

Question: 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

In [10]:

```
num = input("Enter numbers seperated only by a comma :")
numbers_split = num.split(',')

number_tuple = tuple(numbers_split)

print(numbers_split)
print(number_tuple)
```

```
Enter numbers seperated only by a comma :34,67,55,33,12,98
['34', '67', '55', '33', '12', '98']
('34', '67', '55', '33', '12', '98')
```

Question 5

Level 1

Question: 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

In [ ]:

```
class myclass():
    def __init__(self):
        self.str1=''
    def getString(self):
        self.str1=input('Enter String:')
    def printString(self):
        print(self.str1.upper())

str1=myclass()
str1.getString()
str1.printString()
```

Question 6

Level 2

Question: 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

Hints: If the output received is in decimal form, it should be rounded off to its nearest value (for example, if the output received is 26.0, it should be printed as 26) In case of input data being supplied to the question, it should be assumed to be a console input.

In [16]:

```
import math
d = 100,150,180
def Q(d):
    c,h = 50,30
    Q = math.sqrt((2*c*d)/h)
    return Q
for i in d:
    ans = Q(i)
    print(int(ans), end = ', ')
```

```
18, 22, 24,
```

Question 7

Level 2

Question: 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]]

Hints: Note: In case of input data being supplied to the question, it should be assumed to be a console input in a comma-separated form.

In [17]:

```
x,y = 3,5
multi_list = [[0 for j in range(y)] for i in range(x)]

for i in range(x):
    for j in range(y):
        multi_list[i][j]= i*j
print(multi_list)
```

```
[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]
```

Question 8

Level 2

Question: 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

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

In [22]:

```
phrase = input("Input comma seperated sequence of words: ")

phrase_list = phrase.split(",")
phrase_list.sort()
print((', ').join(phrase_list))
```

```
Input comma seperated sequence of words: without,hello,bag,world
bag, hello, without, world
```

Question 9

Level 2

Question£º 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

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

In [24]:

```
y = str(input())
x = y.upper()
print(x)
```

```
Hello world Practice makes perfect
HELLO WORLD PRACTICE MAKES PERFECT
```

Question 10

Level 2

Question: 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

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. We use set container to remove duplicated data automatically and then use sorted() to sort the data.

In [25]:

```
sentence = "hello world and practice makes perfect and hello world again"
spl = sentence.split()
spl = set(spl)
spl = sorted(spl)
for i in spl:
    print(i, end = ' ')
```

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

Question 11

Level 2

Question: 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.

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

In [26]:

```
items = []
num = [x for x in input().split(',')]
for p in num:
    x = int(p, 2)
    if not x%5:
        items.append(p)
print(','.join(items))
```

```
0100,0011,1010,1001
1010
```

Question 12

Level 2

Question: 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.

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

In [27]:

```
ans = []
k = 0
for i in range(1000,3001):
    if i%2 == 0:
        print(i, end=',')
```

```
1000,1002,1004,1006,1008,1010,1012,1014,1016,1018,1020,1022,1024,1026,1028,1
030,1032,1034,1036,1038,1040,1042,1044,1046,1048,1050,1052,1054,1056,1058,10
60,1062,1064,1066,1068,1070,1072,1074,1076,1078,1080,1082,1084,1086,1088,109
0,1092,1094,1096,1098,1100,1102,1104,1106,1108,1110,1112,1114,1116,1118,112
0,1122,1124,1126,1128,1130,1132,1134,1136,1138,1140,1142,1144,1146,1148,115
0,1152,1154,1156,1158,1160,1162,1164,1166,1168,1170,1172,1174,1176,1178,118
0,1182,1184,1186,1188,1190,1192,1194,1196,1198,1200,1202,1204,1206,1208,121
0,1212,1214,1216,1218,1220,1222,1224,1226,1228,1230,1232,1234,1236,1238,124
0,1242,1244,1246,1248,1250,1252,1254,1256,1258,1260,1262,1264,1266,1268,127
0,1272,1274,1276,1278,1280,1282,1284,1286,1288,1290,1292,1294,1296,1298,130
0,1302,1304,1306,1308,1310,1312,1314,1316,1318,1320,1322,1324,1326,1328,133
0,1332,1334,1336,1338,1340,1342,1344,1346,1348,1350,1352,1354,1356,1358,136
0,1362,1364,1366,1368,1370,1372,1374,1376,1378,1380,1382,1384,1386,1388,139
0,1392,1394,1396,1398,1400,1402,1404,1406,1408,1410,1412,1414,1416,1418,142
0,1422,1424,1426,1428,1430,1432,1434,1436,1438,1440,1442,1444,1446,1448,145
0,1452,1454,1456,1458,1460,1462,1464,1466,1468,1470,1472,1474,1476,1478,148
0,1482,1484,1486,1488,1490,1492,1494,1496,1498,1500,1502,1504,1506,1508,151
0,1512,1514,1516,1518,1520,1522,1524,1526,1528,1530,1532,1534,1536,1538,154
0,1542,1544,1546,1548,1550,1552,1554,1556,1558,1560,1562,1564,1566,1568,157
0,1572,1574,1576,1578,1580,1582,1584,1586,1588,1590,1592,1594,1596,1598,160
0,1602,1604,1606,1608,1610,1612,1614,1616,1618,1620,1622,1624,1626,1628,163
0,1632,1634,1636,1638,1640,1642,1644,1646,1648,1650,1652,1654,1656,1658,166
0,1662,1664,1666,1668,1670,1672,1674,1676,1678,1680,1682,1684,1686,1688,169
0,1692,1694,1696,1698,1700,1702,1704,1706,1708,1710,1712,1714,1716,1718,172
0,1722,1724,1726,1728,1730,1732,1734,1736,1738,1740,1742,1744,1746,1748,175
0,1752,1754,1756,1758,1760,1762,1764,1766,1768,1770,1772,1774,1776,1778,178
0,1782,1784,1786,1788,1790,1792,1794,1796,1798,1800,1802,1804,1806,1808,181
0,1812,1814,1816,1818,1820,1822,1824,1826,1828,1830,1832,1834,1836,1838,184
0,1842,1844,1846,1848,1850,1852,1854,1856,1858,1860,1862,1864,1866,1868,187
0,1872,1874,1876,1878,1880,1882,1884,1886,1888,1890,1892,1894,1896,1898,190
0,1902,1904,1906,1908,1910,1912,1914,1916,1918,1920,1922,1924,1926,1928,193
0,1932,1934,1936,1938,1940,1942,1944,1946,1948,1950,1952,1954,1956,1958,196
0,1962,1964,1966,1968,1970,1972,1974,1976,1978,1980,1982,1984,1986,1988,199
0,1992,1994,1996,1998,2000,2002,2004,2006,2008,2010,2012,2014,2016,2018,202
0,2022,2024,2026,2028,2030,2032,2034,2036,2038,2040,2042,2044,2046,2048,205
0,2052,2054,2056,2058,2060,2062,2064,2066,2068,2070,2072,2074,2076,2078,208
0,2082,2084,2086,2088,2090,2092,2094,2096,2098,2100,2102,2104,2106,2108,211
0,2112,2114,2116,2118,2120,2122,2124,2126,2128,2130,2132,2134,2136,2138,214
0,2142,2144,2146,2148,2150,2152,2154,2156,2158,2160,2162,2164,2166,2168,217
0,2172,2174,2176,2178,2180,2182,2184,2186,2188,2190,2192,2194,2196,2198,220
0,2202,2204,2206,2208,2210,2212,2214,2216,2218,2220,2222,2224,2226,2228,223
0,2232,2234,2236,2238,2240,2242,2244,2246,2248,2250,2252,2254,2256,2258,226
0,2262,2264,2266,2268,2270,2272,2274,2276,2278,2280,2282,2284,2286,2288,229
0,2292,2294,2296,2298,2300,2302,2304,2306,2308,2310,2312,2314,2316,2318,232
0,2322,2324,2326,2328,2330,2332,2334,2336,2338,2340,2342,2344,2346,2348,235
0,2352,2354,2356,2358,2360,2362,2364,2366,2368,2370,2372,2374,2376,2378,238
0,2382,2384,2386,2388,2390,2392,2394,2396,2398,2400,2402,2404,2406,2408,241
0,2412,2414,2416,2418,2420,2422,2424,2426,2428,2430,2432,2434,2436,2438,244
0,2442,2444,2446,2448,2450,2452,2454,2456,2458,2460,2462,2464,2466,2468,247
0,2472,2474,2476,2478,2480,2482,2484,2486,2488,2490,2492,2494,2496,2498,250
0,2502,2504,2506,2508,2510,2512,2514,2516,2518,2520,2522,2524,2526,2528,253
```

```
0,2532,2534,2536,2538,2540,2542,2544,2546,2548,2550,2552,2554,2556,2558,256
0,2562,2564,2566,2568,2570,2572,2574,2576,2578,2580,2582,2584,2586,2588,259
0,2592,2594,2596,2598,2600,2602,2604,2606,2608,2610,2612,2614,2616,2618,262
0,2622,2624,2626,2628,2630,2632,2634,2636,2638,2640,2642,2644,2646,2648,265
0,2652,2654,2656,2658,2660,2662,2664,2666,2668,2670,2672,2674,2676,2678,268
0,2682,2684,2686,2688,2690,2692,2694,2696,2698,2700,2702,2704,2706,2708,271
0,2712,2714,2716,2718,2720,2722,2724,2726,2728,2730,2732,2734,2736,2738,274
0,2742,2744,2746,2748,2750,2752,2754,2756,2758,2760,2762,2764,2766,2768,277
0,2772,2774,2776,2778,2780,2782,2784,2786,2788,2790,2792,2794,2796,2798,280
0,2802,2804,2806,2808,2810,2812,2814,2816,2818,2820,2822,2824,2826,2828,283
0,2832,2834,2836,2838,2840,2842,2844,2846,2848,2850,2852,2854,2856,2858,286
0,2862,2864,2866,2868,2870,2872,2874,2876,2878,2880,2882,2884,2886,2888,289
0,2892,2894,2896,2898,2900,2902,2904,2906,2908,2910,2912,2914,2916,2918,292
0,2922,2924,2926,2928,2930,2932,2934,2936,2938,2940,2942,2944,2946,2948,295
0,2952,2954,2956,2958,2960,2962,2964,2966,2968,2970,2972,2974,2976,2978,298
0,2982,2984,2986,2988,2990,2992,2994,2996,2998,3000,
```

Question 13

Level 2

Question: 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

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

In [30]:

```
sentence = str(input())
sentence = list(sentence)
letter = 0
digit = 0
for i in range(len(sentence)):
    if sentence[i].isnumeric():
        digit += 1
    else:
        letter += 1
print('LETTERS %d \nDIGITS %d' %(letter, digit))
```

```
hello world! 123
LETTERS 13
DIGITS 3
```

Question 14

Level 2

Question: 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

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

In [32]:

```
sentence = str(input())
sentence = list(sentence)
upper = 0
lower = 0
for i in range(len(sentence)):
    if sentence[i].isupper():
        upper += 1
    if sentence[i].islower():
        lower += 1
print('UPPER CASE %d \nLOWER CASE %d' %(upper, lower))
```

```
Hello world!
UPPER CASE 1
LOWER CASE 9
```

Question 15

Level 2

Question: 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

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

In [34]:

```
n=input('')
result=0
for i in range(1,5):
  a=n*i
  new=int(a)
  result=result+new
print(result)
```

```
9
11106
```

Question 16

Level 2

Question: 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

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

In [36]:

```
lis = [x for x in input('').split(',')]
for i in range(len(lis)):
  if int(lis[i])%2 ==1:
    print(lis[i],end = ',')
```

```
1,2,3,4,5,6,7,8,9
1,3,5,7,9 ,
```

Question 17

Level 2

Question: Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following: D 100 W 200

D means deposit while W means withdrawal. Suppose the following input is supplied to the program: D 300 D 300 W 200 D 100 Then, the output should be: 500

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

Solution:

In [39]:

```
total = 0
while 1:
    money = input()
    if money == "":
        break
    else:
        money = money.split(" ")
        if money[0] == "D":
            total = total + int(money[1])
        elif money[0] == "W":
            total = total - int(money[1])

print(total)
```

```
D 300 D 300 W 200 D 100

300
```

Question 18

Level 3

Question: A website requires the users to input username and password to register. Write a program to check the validity of password input by users. Following are the criteria for checking the password:

1. At least 1 letter between [a-z]
2. At least 1 number between [0-9]
3. At least 1 letter between [A-Z]
4. At least 1 character from [$#@]

5. Minimum length of transaction password: 6
6. Maximum length of transaction password: 12 Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma. Example If the following passwords are given as input to the program: ABd1234@1,a F1#,2w3E*,2We3345 Then, the output of the program should be: ABd1234@1

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

In [41]:

```
s=input("enter sequence of comma separated passwords :")
import re
l=s.split(",")
for i in l:
  if(len(i)>=6 and len(i)<=12):
    if re.search("([a-z])+", i):
      if re.search("([A-Z])+", i):
        if re.search("([0-9])+", i):
          if re.search("([#,@,$])+", i):
            print(i)
```

```
enter sequence of comma separated passwords :ABd1234@1,a F1#,2w3E*,2We3345
ABd1234@1
```

Question 19

Level 3

Question: You are required to write a program to sort the (name, age, height) tuples by ascending order where name is string, age and height are numbers. The tuples are input by console. The sort criteria is: 1: Sort based on name; 2: Then sort based on age; 3: Then sort by score. The priority is that name > age > score. If the following tuples are given as input to the program: Tom,19,80 John,20,90 Jony,17,91 Jony,17,93 Json,21,85 Then, the output of the program should be: [('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json', '21', '85'), ('Tom', '19', '80')]

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

In [43]:

```
s = input('')
lst = [tuple(x.split(',')) for x in s.split()]

print(sorted(lst, key=lambda x: (x[0], x[1], x[2])))
```

```
Tom,19,80 John,20,90 Jony,17,91 Jony,17,93 Json,21,85
[('John', '20', '90'), ('Jony', '17', '91'), ('Jony', '17', '93'), ('Json',
'21', '85'), ('Tom', '19', '80')]
```

Question 20

Level 3

Question: Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

Hints: Consider use yield

Solution:

In [47]:

```
def div7(n):
    for i in range(1,n+1):
        if i % 7 == 0:
            yield i
print(div7(21))
```

```
<generator object div7 at 0x00000218CD2F3430>
```

Question 21

Level 3

Question
A robot moves in a plane starting from the original point (0,0). The robot can move toward UP, DOWN, LEFT and RIGHT with a given steps. The trace of robot movement is shown as the following:
UP 5
DOWN 3
LEFT 3
RIGHT 2
¡•
The numbers after the direction are steps. Please write a program to compute the distance from current position after a sequence of movement and original point. If the distance is a float, then just print the nearest integer.
Example:
If the following tuples are given as input to the program:
UP 5
DOWN 3
LEFT 3
RIGHT 2
Then, the output of the program should be:
2

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

In [50]:

```
x = ['UP ','DOWN ',"LEFT ",'RIGHT ']
y = list(range(len(x)))
for i in range(len(x)):
  y[i] = input(x[i])
  i += 1
vertical = abs(int(y[0])-int(y[1]))
horizontal = abs(int(y[2])-int(y[3]))
if vertical > horizontal:
  print(vertical)
else:
  print(horizontal)
```

```
UP 5
DOWN 3
LEFT 3
RIGHT 2
2
```

Question 22

Level 3

Question: Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically. Suppose the following input is supplied to the program: New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3. Then, the output should be: 2:2 3.:1 3?:1 New:1 Python:5 Read:1 and:1 between:1 choosing:1 or:2 to:1

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

In [52]:

```
def frequency(input):
    freq = {}

    for word in input.split():
        freq[word] = freq.get(word, 0) + 1

    words = list(freq.keys())
    words.sort()

    for w in words:
        print(f'{w}:{freq[w]}')

x = input()
print(frequency(x))
```

```
New to Python or choosing between Python 2 and Python 3? Read Python 2 or Py
thon 3
2:2
3:1
3?:1
New:1
Python:5
Read:1
and:1
between:1
choosing:1
or:2
to:1
None
```

Question 23

level 1

Question: Write a method which can calculate square value of number

Hints: Using the ** operator

In [55]:

```
def square(x):
  y = x**2
  return y
x = int(input())
print(square(x))
```

```
5
25
```

Question 24

Level 1

Question:

Python has many built-in functions, and if you do not know how to use it, you can read document online or find some books. But Python has a built-in document function for every built-in functions.

Please write a program to print some Python built-in functions documents, such as abs(), int(), raw_input()

And add document for your own function Hints: The built-in document method is **doc**

In [59]:

```
print(abs.__doc__)
print(int.__doc__)
#print(raw_input.__doc__) #python doesnt know about the raw_input
```

```
Return the absolute value of the argument.
int([x]) -> integer
int(x, base=10) -> integer

Convert a number or string to an integer, or return 0 if no arguments
are given.  If x is a number, return x.__int__().  For floating point
numbers, this truncates towards zero.

If x is not a number or if base is given, then x must be a string,
bytes, or bytearray instance representing an integer literal in the
given base.  The literal can be preceded by '+' or '-' and be surrounded
by whitespace.  The base defaults to 10.  Valid bases are 0 and 2-36.
Base 0 means to interpret the base from the string as an integer literal.
>>> int('0b100', base=0)
4
```

Question 25

Level 1

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

Hints: Define a instance parameter, need add it in **init** method You can init a object with construct parameter or set the value later

In [63]:

```
class Point:
    def __init__(self):
        self.x = 0
        self.y = 0
```

Question 26:

Define a function which can compute the sum of two numbers.

Hints: Define a function with two numbers as arguments. You can compute the sum in the function and return the value.

In [65]:

```
def sum(x,y):
  return x+y
print(sum(3,5))
```

```
8
```

Question 27

Define a function that can convert a integer into a string and print it in console.

Hints:

Use str() to convert a number to string.

In [67]:

```
def int_str(x):
  return str(x)
print(type(int_str(5)))
```

```
<class 'str'>
```

Question 28

Define a function that can convert a integer into a string and print it in console.

Hints:

Use str() to convert a number to string.

In [68]:

```
def int_str(x):
  return str(x)
print(type(int_str(5)))
```

```
<class 'str'>
```

Question 29

Define a function that can receive two integral numbers in string form and compute their sum and then print it in console.

Hints:

Use int() to convert a string to integer.

In [69]:

```
def sum(a,b):
    s = int(a) + int(b)
    return s
num1, num2 = "2","5"
print(sum(num1,num2))
```

```
7
```

Question 30

Define a function that can accept two strings as input and concatenate them and then print it in console.

Hints:

Use + to concatenate the strings

In [70]:

```
def concat(a,b):
  return a + b
print(concat("Hello","World"))
```

```
HelloWorld
```

Question 31

Define a function that can accept two strings as input and print the string with maximum length in console. If two strings have the same length, then the function should print al l strings line by line.

Hints:

Use len() function to get the length of a string

In [71]:

```
def maxPrint(a,b):
  if len(a)>len(b):
    print(a)
  if len(b)>len(a):
    print(b)
  else:
    print(a+"\n"+b)
maxPrint("Hello","World")
```

```
Hello
World
```

Question 32

Define a function that can accept an integer number as input and print the "It is an even number" if the number is even, otherwise print "It is an odd number".

Hints:

Use % operator to check if a number is even or odd.

In [72]:

```
def oddEven(a):
  if int(a)%2 == 0:
    print("It is an even number")
  else:
    print("It is an odd number")
oddEven(5)
oddEven(8)
```

```
It is an odd number
It is an even number
```

Question 33

Define a function which can print a dictionary where the keys are numbers between 1 and 3 (both included) and the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use ** operator to get power of a number.

In [75]:

```
def squareDict1():
  d = dict()
  for x in range(1,4):
      d[x]=x**2
  return d
print(squareDict1())
```

```
{1: 1, 2: 4, 3: 9}
```

Question 34

Define a function which can print a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use ** operator to get power of a number. Use range() for loops.

In [76]:

```
def squareDict2():
  d = dict()
  for x in range(1,21):
      d[x]=x**2
  return d
print(squareDict2())
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 12
1, 12: 144, 13: 169, 14: 196, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361, 2
0: 400}
```

Question 35

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.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use ** operator to get power of a number. Use range() for loops. Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.

In [77]:

```
def squareDict3():
  d = dict()
  for x in range(1,21):
      d[x]=x**2
  return d.values()
print(squareDict3())
```

```
dict_values([1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225,
256, 289, 324, 361, 400])
```

Question 36

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.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use ** operator to get power of a number. Use range() for loops. Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.

In [78]:

```
def squareDict4():
  d = dict()
  for x in range(1,21):
      d[x]=x**2
  return d.keys()
print(squareDict3())
```

```
dict_values([1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225,
256, 289, 324, 361, 400])
```

Question 37

Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (both included).

Hints:

Use ** operator to get power of a number. Use range() for loops. Use list.append() to add values into a list.

In [79]:

```
def lis():
  k = list()
  for i in range(1,21):
    k.append(i**2)
  return k
print(lis())
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 32
4, 361, 400]
```

Question 38

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.

Hints:

Use ** operator to get power of a number. Use range() for loops. Use list.append() to add values into a list. Use [n1:n2] to slice a list

In [80]:

```
def lis1():
  k = list()
  for i in range(1,21):
    k.append(i**2)
  return k[:5]
print(lis1())
```

```
[1, 4, 9, 16, 25]
```

Question 39

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 last 5 elements in the list.

Hints:

Use ** operator to get power of a number. Use range() for loops. Use list.append() to add values into a list. Use [n1:n2] to slice a list

In [81]:

```
def lis2():
  k = list()
  for i in range(1,21):
    k.append(i**2)
  return k[-5:]
print(lis2())
```

```
[256, 289, 324, 361, 400]
```

Question 40

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 all values except the first 5 elements in the list.

Hints:

Use ** operator to get power of a number. Use range() for loops. Use list.append() to add values into a list. Use [n1:n2] to slice a list

In [82]:

```
def lis3():
  k = list()
  for i in range(1,21):
    k.append(i**2)
  return k[5:]
print(lis3())
```

```
[36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400]
```

Question 41

Define a function which can generate and print a tuple where the value are square of numbers between 1 and 20 (both included).

Hints:

Use ** operator to get power of a number. Use range() for loops. Use list.append() to add values into a list. Use tuple() to get a tuple from a list.

In [83]:

```
def tup():
  x = list()
  for i in range(1,21):
    x.append(i**2)
  return tuple(x)
print(tup())
```

```
(1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 32
4, 361, 400)
```

Question 42

With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.

Hints:

Use [n1:n2] notation to get a slice from a tuple.

In [84]:

```
giv_tuple = (1,2,3,4,5,6,7,8,9,10)
print(giv_tuple[0:5],end = '')
print(giv_tuple[5:])
```

```
(1, 2, 3, 4, 5)(6, 7, 8, 9, 10)
```

Question 43

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).

Hints:

Use "for" to iterate the tuple Use tuple() to generate a tuple from a list.

In [85]:

```
giv_tuple = (1,2,3,4,5,6,7,8,9,10)
x = list(giv_tuple)
k = list()
for i in range(len(x)):
  if x[i]%2 == 0:
    k.append(x[i])
print(tuple(k))
```

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

Question 44

Write a program which accepts a string as input to print "Yes" if the string is "yes" or "YES" or "Yes", otherwise print "No".

Hints:

Use if statement to judge condition.

In [86]:

```
x = input('')
if x == "Yes" or x == 'yes' or x == 'YES':
  print('Yes')
else: print('No')
```

```
yes
Yes
```

Question 45

Write a program which can filter even numbers in a list by using filter function. The list is: [1,2,3,4,5,6,7,8,9,10].

Hints:

Use filter() to filter some elements in a list. Use lambda to define anonymous functions

In [87]:

```
lis = [1,2,3,4,5,6,7,8,9,10]
k = list(filter(lambda x: x%2==0, lis))
print(k)
```

```
[2, 4, 6, 8, 10]
```

Question 46

Write a program which can map() to make a list whose elements are square of elements in [1,2,3,4,5,6,7,8,9,10].

Hints Use map() to generate a list. Use lambda to define anonymous functions.

In [88]:

```
lis = [1,2,3,4,5,6,7,8,9,10]
result = list(map(lambda x: x**2, lis))
print(result)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Question 47

Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10].

Hints Use map() to generate a list. Use filter() to filter elements of a list. Use lambda to define anonymous functions.

In [89]:

```
lis = [1,2,3,4,5,6,7,8,9,10]
even_sq = (map(lambda x: x**2, filter(lambda   x: x%2==0, lis)))
print(list(even_sq))
```

```
[4, 16, 36, 64, 100]
```

Question 48

Write a program which can filter() to make a list whose elements are even number between 1 and 20 (both included).

Hints:

Use filter() to filter elements of a list. Use lambda to define anonymous functions.

In [90]:

```
even = list(filter(lambda   x: x%2==0, [x for x in range(1,21)]))
print(even)
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

Question 49

Write a program which can map() to make a list whose elements are square of numbers between 1 and 20 (both included).

Hints Use map() to generate a list. Use lambda to define anonymous functions.

In [91]:

```
sq = map(lambda x: x**2, [x for x in range(1,21)])
print(list(sq))
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 32
4, 361, 400]
```

Question 50

Define a class named American which has a static method called printNationality.

Hints: Use @staticmethod decorator to define class static method.

In [92]:

```
class American(object):
    @staticmethod
    def printNationality():
        print("America")

anAmerican = American()
anAmerican.printNationality()
American.printNationality()
```

```
America
America
```

Question 51

Define a class named American and its subclass NewYorker.

Hints:

Use class Subclass(ParentClass) to define a subclass.

In [93]:

```
class American(object):
    pass

class NewYorker(American):
    pass

anAmerican = American()
aNewYorker = NewYorker()
print(anAmerican)
print(aNewYorker)
```

```
<__main__.American object at 0x00000218CD1CAC40>
<__main__.NewYorker object at 0x00000218CD1C1E50>
```

Question 52

Define a class named Circle which can be constructed by a radius. The Circle class has a method which can compute the area.

Hints:

Use def methodName(self) to define a method.

In [94]:

```
class circle():
  def __init__(self, r):
    self.radius = r
  def area(self):
    return self.radius**2*3.14
c = circle(8)
c.area()
```

Out[94]:

```
200.96
```

Question 53

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.

Hints:

Use def methodName(self) to define a method.

In [95]:

```
class rectArea():
  def __init__(self,x,y):
    self.length = x
    self.width = y
  def area(self):
    return self.length*self.width
b = rectArea(3,4)
b.area()
```

Out[95]:

```
12
```

Question 54

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.

Hints:

To override a method in super class, we can define a method with the same name in the super class.

In [96]:

```
class Shape(object):
    def __init__(self):
        pass
    def area(self):
        return 0
class Square(Shape):
    def __init__(self, l):
        Shape.__init__(self)
        self.length = l
    def area(self):
        return self.length*self.length
Square= Square(3)
print(Square.area())
```

```
9
```

Question 55

Please raise a RuntimeError exception.

Hints:

Use raise() to raise an exception.

In [100]:

```
raise RuntimeError('something is wrong')
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11904\1474207836.py in <module>
----> 1 raise RuntimeError('something is wrong')

RuntimeError: something is wrong
```

Question 56

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

Hints:

Use try/except to catch exceptions.

In [101]:

```
def divide():
    return 5/0

try:
    divide()
except ZeroDivisionError as ze:
    print("Number divide by 0")
except:
    print("Any other exception")
```

```
Number divide by 0
```

Question 57

Define a custom exception class which takes a string message as attribute.

Hints:

To define a custom exception, we need to define a class inherited from Exception.

In [103]:

```
class customError(Exception):
  def __init__(self, *args):
    if args:
      self.message = args[0]
    else:
      self.message = None
  def __str__(self):
    print('Calling str')
    if self.message:
      return 'MyCustomError, {0}'.format(self.message)
    else:
      return 'MyCustomError has been raised'

raise customError('We customized the Error')
```

```
Calling str

---------------------------------------------------------------------------
customError                               Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11904\2221778922.py in <module>
     12       return 'MyCustomError has been raised'
     13
---> 14 raise customError('We customized the Error')

customError: MyCustomError, We customized the Error


Calling str
Calling str
```

Question 58

Assuming that we have some email addresses in the "username@companyname.com (mailto:username@companyname.com)" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

Example: If the following email address is given as input to the program:

john@google.com (mailto:john@google.com)

Then, the output of the program should be:

john

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

Hints:

Use \w to match letters.

In [104]:

```
import re
email = input('')
pat2 = "(\w+)@(\w+)\.(com)"
r2 = re.match(pat2,email)
print(r2.group(1))
```

```
john@google.com
john
```

Question 59

Assuming that we have some email addresses in the "username@companyname.com (mailto:username@companyname.com)" format, please write program to print the company name of a given email address. Both user names and company names are composed of letters only.

Example: If the following email address is given as input to the program:

john@google.com (mailto:john@google.com)

Then, the output of the program should be:

google

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

Hints:

Use \w to match letters.

In [105]:

```
import re
email = input('')
pat2 = "(\w+)@(\w+)\.(com)"
r2 = re.match(pat2,email)
print(r2.group(2))
```

```
john@google.com
google
```

Question 60

Write a program which accepts a sequence of words separated by whitespace as input to print the words composed of digits only.

Example: If the following words is given as input to the program:

2 cats and 3 dogs.

Then, the output of the program should be:

['2', '3']

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

Hints:

Use re.findall() to find all substring using regex.

In [106]:

```
import re
s = input()
print(re.findall("\d+",s))
```

```
2 cats and 3 dogs.
['2', '3']
```

In [ ]:

```

```