In [ ]:

```
# LAB 1
```

In [8]:

```
# QUESTION 1
# Given 2 lists of numbers, create a new list such that it contains odd numbers
 from list1 and even from list2
```

In [136]:

```
def genList(list1, list2):
    list3 = []
    for i in range(len(list1)):
        if(list1[i]%2==1):
            list3.append(list1[i])
    for i in range(len(list2)):
        if(list2[i]%2==0):
            list3.append(list2[i])
    return list3

list1 = [1,2,3,4,5]
list2 = [2,4,5,6,7,8]
print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
print(genList(list1,list2))
```

```
OUTPUT
------
Nilesh Jain - 180953226
[1, 3, 5, 2, 4, 6, 8]
```

In [ ]:

```
# QUESTION 2
# Write a python program to display the pattern
```

In [137]:

```
def genPattern(n):
    count = 1
    rows = 0
    while(rows<n):
        for i in range(n):
            rows = rows+1
            for j in range(i+1):
                if(j==i):
                    print(count,end = "\n")
                else:
                    print(count,end = " ")
                count = count+1

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
genPattern(5)
```

```
OUTPUT
------
Nilesh Jain - 180953226
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

In [ ]:

```
# QUESTION 3
# Write a program to read n strings, display those with odd length and count number of strings with same first and
# last character, and length > 2
```

In [138]:

```
def dispOdd(listStrings):
    oddString = []
    for string in listStrings:
        if(len(string)%2==1):
            oddString.append(string)
    return oddString

def sameString(listStrings):
    count = 0
    for string in listStrings:
        if(len(string)>=2):
            if(string[0]==string[len(string)-1]):
                count = count+1
    return count

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
strings = ["test", "Nilesh", "six", "seven", "odd", "even", "racecar"]
print("Strings with odd length -", dispOdd(strings))
print("Number of strings with same starting and ending character -", sameString(
strings))
```

```
OUTPUT
------
Nilesh Jain - 180953226
Strings with odd length - ['six', 'seven', 'odd', 'racecar']
Number of strings with same starting and ending character - 2
```

In [ ]:

```
```

In [ ]:

```
# LAB 2
```

In [ ]:

```
# QUESTION 1
# Count the number of words in a given sentence using dictionaries
```

In [139]:

```
def countWords(sentence):
    dict = {}
    words = sentence.split(" ")
    count = 0
    for word in words:
        dict[word] = 0
    for word in words:
        dict[word] += 1
        count = count+1
    print(dict)
    return count

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
print("The total number of words are:",countWords("Hello there, this is Nilesh.
 Hello World!"))
```

```
OUTPUT
------
Nilesh Jain - 180953226
{'Hello': 2, 'there,': 1, 'this': 1, 'is': 1, 'Nilesh.': 1, 'Worl
d!': 1}
The total number of words are: 7
```

In [ ]:

```
# QUESTION 2
# Write a program to read the order of 2 matrices, store nonzero values, add and
display resultant matrix
```

In [140]:

```
def addMatrix(dict1, dict2, rows,cols):
    sumMatrix = []
    for i in range(rows):
        sumRow = []
        for j in range(cols):
            sumRow.append(dict1[i][j]+dict2[i][j])
        sumMatrix.append(sumRow)
    return sumMatrix

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
m1 = int(input('Enter number of rows in matrix 1:'))
n1 = int(input('Enter number of columns in matrix 1:'))
m2 = int(input('Enter number of rows in matrix 2:'))
n2 = int(input('Enter number of columns in matrix 2:'))
dict1 = {}
dict2 = {}
if(m1!=m2 or n1!=n2):
    print('Invalid. Enter matrices of same dimension!')
else:
    for i in range(m1):
        row = []
        for j in range(n1):
            val = int(input('Enter non zero value:'))
            row.append(val)
        dict1[i] = row

    for i in range(m2):
        row = []
        for j in range(n2):
            val = int(input('Enter non zero value:'))
            row.append(val)
        dict2[i] = row

    sumMatrix = addMatrix(dict1,dict2,m1,n1)
    print("\n")
    for i in range(m1):
        for j in range(n1):
            if(j==n1-1):
                print(sumMatrix[i][j],end="\n")
            else:
                print(sumMatrix[i][j], end=" ")
```

```
OUTPUT
------
Nilesh Jain - 180953226
Enter number of rows in matrix 1:2
Enter number of columns in matrix 1:3
Enter number of rows in matrix 2:2
Enter number of columns in matrix 2:3
Enter non zero value:1
Enter non zero value:2
Enter non zero value:3
Enter non zero value:4
Enter non zero value:5
Enter non zero value:6
Enter non zero value:1
Enter non zero value:2
Enter non zero value:3
Enter non zero value:4
Enter non zero value:5
Enter non zero value:6


2 4 6
8 10 12
```

In [ ]:

```
# QUESTION 3
```

In [135]:

```
import random

print('OUTPUT')
print('------')
print('Nilesh Jain - 180953226')
print("\n")
dict = {}
n = int(input("Enter number of values:"))
for i in range(n):
    dict[random.randint(0,100)] = input("Enter value:")

sum=0
count=0
string = ""
for key,value in dict.items():
    if(value.isdigit()):
        sum+=int(value)
        count+=1
    else:
        string+=value

print("\n")
print('Average is:', sum/count)
print('Concatenated string is:', string)

found = False
print("\n")
search = input('Enter value to be searched:')
if(search.isdigit()):
    print('Please enter string')
else:
    for key,value in dict.items():
        if(value==search):
            found=True
            print('Element', value, "found at key", key)
    if(found==False):
        print("Element not found!")

print('Elements with only special characters:')
for key, value in dict.items():
    special = True
    for i in range(len(value)):
        if(value[i].isalpha() or value[i].isdigit()):
            special = False
            break
    if(special==True):
        print(value)
```

```
OUTPUT
------
Nilesh Jain - 180953226


Enter number of values:6
Enter value:1
Enter value:2
Enter value:a
Enter value:b
Enter value:$&^$
Enter value:$123ab


Average is: 1.5
Concatenated string is: ab$&^$$123ab


Enter value to be searched:a
Element a found at key 10
Elements with only special characters:
$&^$
```