Task 13

Registration ID : SIRSS1038

Leela Satya Kartheek Raja

Q1) Given a square matrix, turn it by 90 degrees in a clockwise direction without using anyextra space.

In [2]:

```
def Rotate90(x):
    n= len(x[0])
    for i in range(n//2):
        for j in range(i,n-i-1):
            temp=x[i][j]

            x[i][j]=x[n-1-j][i]

            x[n-1-j][i]=x[n-1-i][n-1-j]

            x[n-1-i][n-1-j]=x[j][n-1-i]

            x[j][n-1-i]=temp

def printM(x):
    n=len(x[0])
    for i in range(n):
        print(x[i])
```

In [3]:

```
x=[[1,2,3],[4,5,6],[7,8,9]]

Rotate90(x)
printM(x)
```

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

Q2) Given a string S, check if it is palindrome or not.

In [4]:

```
def Palind(x):
    for i in range(0,int(len(x)/2)):
        if x[i]!=x[len(x)-i-1]:
            return False
        return True
string=input("")
result=Palind(string)

if(result):
    print("yes its a Palindrome")
else:
    print("Not a palindrome")
```

```
123
Not a palindrome
```

Q3) Given an expression string x. Examine whether the pairs and the orders of “{“,”}”,”(“,”)”,”[“,”]” are correct in exp. For example, the function should return 'true' for exp = “[()]{}{[(()()}” and 'false' for exp = “[(])”.

In [5]:

```
def balbracket(expr):
    l = []

    for i in expr:
        if i in ["(", "{", "["]:

            l.append(i)
        else:

            if not l:
                return False
            cc = l.pop()
            if cc == '(':
                if i != ")":
                    return False
            if cc == '{':
                if i != "}":
                    return False
            if cc == '[':
                if i != "]":
                    return False

    if l:
        return False
    return True
```

In [6]:

```
if __name__ == "__main__":
    expr = input("")

    if balbracket(expr):
        print("Balanced")
    else:
        print("Not Balanced")
```

```
[({}}}
Not Balanced
```

Q4) Problem Statement: Selection of MPCS exams include a fitness test which is conducted on ground. There will be a batch of 3 trainees, appearing for running test in track for 3 rounds. You need to record their oxygen level after every round. After the trainees are finished with all rounds, calculate for each trainee his average oxygen level over the 3 rounds and select one with highest oxygen level as the most fit trainee. Thus, if more than one trainee attains the same highest average level, then all need to be selected.

Display the most fit trainee (or trainees) and the highest average oxygen level.

Note:

The oxygen value entered should not be accepted if it is not in the range between 1 and 100. Moreover, if the calculated maximum average oxygen value of trainees is below 70 then declare the trainees as unfit with meaningful message as “All trainees are unfit”. Average Oxygen Values should be rounded.

In [7]:

```
trainee = [[],[],[],[]]
for i in range(3):
    for j in range(3):
        trainee[i].append(int(input()))
        if (trainee[i][-1]) not in range(1,101):
            print("invalid input")

for i in range(3):
    trainee[3].append((trainee[2][i]+trainee[1][i]+trainee[0][i])//3)

maximum = max(trainee[3])

for i in range(3):
    if trainee[3][i] < 70 :
        print("Trainee {0} is unfit".format(i+1))

    elif trainee[3][i] == maximum:
        print("Trainee Number: ",i+1)
```

```
95
98
97
91
95
98
97
95
91
Trainee Number:  2
```

In [ ]:

```

```