INFYTQ FINAL ROUNDQUESTIONS

TEST PATTERN

Objective Question Topics:

- DBMS – 10 ques
- Java/python – 10ques

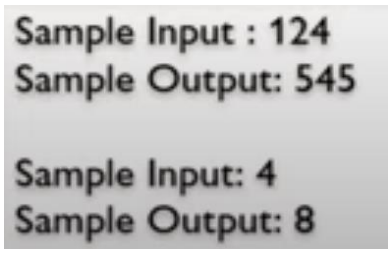
Coding Question Topics:

- Javaor
- Pyhton (2ques)

1.Problemstatement

For a given positive num , identify the palindrome formed by performing following operations.

- Add numbers and itsreverse
- Check whether the sum is palindrome or not. If not, add the sum and itsreverse and repeat the process until a palindrome isobtained



SOLUTION:

```
def isPalindrome(num):
    return str(num)[::-1]==str(num
def rev(num):
    return int(str(num)[::-1])
num=int(input())
while(1):
    num=num+rev(num)
    if(isPalindrome(num)):
        print(num)
        break
```

2.Problem Statement -: Given a m x n matrix inmatrix of positive integers, printan integer outnum based on the belowlogic:

- Identify all possible sets in inmatrix that contain at least four consecutiveelements of the same value val, either horizontally, vertically, ordiagonally
- If only one set of consecutive elements is identified, store the value val inoutnum
- If more than one set of consecutive elements is identified, find the smallestvalue and store it in outnum
- If no set of four consecutive elements of the same value is identifiedeither horizontally, vertically, or diagonally, print-1

Assumption:

- m and n will be greater than3

Input format:

- First line will contain number of rows m ofinmatrix
- The next m lines will contain the elements of inmatrix. Each line will haven elements separated byspace.
- Read the input from the standard inputstream.

Output format:

- Print outnum to the standard outputstream.

- **Sample Input1**
  5
  0 1 6 8 89
  5 6 1 6 89
  6 5 6 1 19
  1 6 6 1 19
  6 3 3 3 39
- **Sample Output1**
  1
- **Explanation1**
  Following elements are present consecutively at least four times: Element 3 horizontally in the 5th row. Element 1 diagonally starting from the 2nd column in the first row. Element 6 diagonally starting from the 4th column in the second row. Element 9 vertically in the 6th column. As element 1 is the smallest value of the four identified sets of consecutive values, the output is 1

CODE:

```
r=int(input())
a=[]
for i in range(r):
 b=list(map(int,input().split()))
 a.append(b)
c=len(a[0])
ans=set()
i=0
while i<r:
 j=0
 while j<(c-3):
  if a[i][j]==a[i][j+1] and a[i][j]==a[i][j+2] and a[i][j]==a[i][j+3]:
   ans.add(a[i][j])
   j+=4
  else:
   j+=1
 i+=1
j=0
while j<c:
 i=0
 while i<(r-3):
  if a[i][j]==a[i+1][j] and a[i][j]==a[i+2][j] and a[i][j]==a[i+3][j]:
   ans.add(a[i][j])
   i+=4
  else:
   i+=1
 j+=1
for i in range(r):
 i1=i
 j=0
 while (i1+3)<r and (j+3)<c:
  if a[i1][j]==a[i1+1][j+1] and a[i1][j]==a[i1+2][j+2] and a[i1][j]==a[i1+3][j+3]:
    i1+=4
    j+=4
    ans.add(a[i1][j])
  else:
```

```
    i1+=1
    j+=1
for j1 in range(1,c):
 j=j1
 i1=0
 while (i1+3)<r and (j+3)<c:
  if a[i1][j]==a[i1+1][j+1] and a[i1][j]==a[i1+2][j+2] and a[i1][j]==a[i1+3][j+3]:
    ans.add(a[i1][j])
    i1+=4
    j+=4
  else:
   i1+=1
   j+=1
for i in range(r):
 i1=i
 j=(c-1)
 while (i1+3)<r and (j-3)>=0:
  if a[i1][j]==a[i1+1][j-1] and a[i1][j]==a[i1+2][j-2] and a[i1][j]==a[i1+3][j-3]:
    i1+=4
    j-=4
    ans.add(a[i1][j])
  else:
   i1+=1
   j-=1
for j1 in range(c-2,-1,-1):
 j=j1
 i1=0
 while (i1+3)<r and (j-3)>=0:
  if a[i1][j]==a[i1+1][j-1] and a[i1][j]==a[i1+2][j-2] and a[i1][j]==a[i1+3][j-3]:
    ans.add(a[i1][j])
    i1+=4
    j-=4
  else:
   i1+=1
   j-=1
if len(ans)>=1:
 print(min(ans))
else:
 print(-1)
```

3.Problem Statement -: Consider an array inarr containing at least two non-zero positive integers ranging between 1 to 300 (inclusive of the boundary values). Divide the integers in inarr into two groups based on the belowrules:

1. Each of the integers should belong to either of the twogroups
2. The total sum of integers in each of the groups must be as nearly equal aspossible
3. The total number of integers between the two groups should not differ bymore than1

Print, outnum1 and outnum2, the sum of the integers of two groups separated by aspace. If outnum1 and outnum2 are not equal, then print the smaller sum followed by the larger sum.

Input Format:

- Read the array inarr elements separated by (‘,’)comma.
- Read the input from the standard inputstream.

Output Format:

- Print outnum1 and outnum2 in the required order separated by aspace.
- Print the output to the standard outputstream.

Sample Test Cases

- **SampleInput**
  87,100,28,67,68,41,67,1

- **SampleOutput**
  229 230

Explanation

For the given input, the two groups that can be formed following the mentioned rulesare:

1. Group 1: 87 100 411
2. Group 2: 28 67 6867

The total sum of integers in each of the groups which is as nearly equal as possible is:

1. Group 1-TotalSum:229
2. Group 2-TotalSum:230

The total number of integers between group 1 and 2 differ by O integer.

SOLUTION:

```
a=list(map(int,input().split(',')))
n1=len(a)
n2=n1//2
su=sum(a)/2
dp=[[-1 for i in range(n2+1)] for j in range(n1)]
def answer(i,n,s):
 if i<(n-1):
  return float('inf')
 if n==0:
  return abs(s)
 else:
  return min(answer(i-1,n-1,s-a[i-1]),answer(i-1,n,s))

k=answer(n1-1,n2,su)
print(int(su-k),int(su+k))
```

```
87,100,28,67,68,41,67,1
229 230
```

4.REVERSE A STRING KEEPING THE SPECIAL CHARACTERS ATTHE SAMEPLACE

Sample input:

#ab$is

Sample output:

#si$ba

CODE:

```
s=input("enter a string")
s1=""
for i in range(len(s)-1,-1,-1):
    if s[i].isalpha():
        s1=s1+s[i]
s1=list(s1)
for i in range(len(s)):
    if s[i].isalpha()==False:
        s1.insert(i,s[i])
print("".join(s1))
```

```
enter a string#ab$is
#si$ba
```