QUESTION 1:

Input : Given a List of numbers separated with comma.
The numbers 5 and 8 are present in the list.
Assume that 8 always comes after 5.
Case 1: num1 -> Add all numbers which do not lie between 5 and 8 in the Input List.
Case 2: num2 -> Numbers formed by concatenating all numbers from 5 to 8 in the list.
Output: Sum of num1 and num2
Example: 3,2,6,5,1,4,8,9
Num1: 3+2+6+9 =20
Num2: 5148
O/p=5148+20 = 5168

Qno : 1 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Question1 {

public static void main(String[] args)throws Exception {

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str[]=br.readLine().split(",");

int num1=0;

String num2="";

int n=str.length;

int arr[]=new int[n];

int fiveIndex = -1;

int eightIndex = -1;

for(int i = 0 ;i < n; i++)//converting string array into Integer array.

{

        arr[i]=Integer.parseInt(str[i]);

        if(arr[i]==5)// finding index of 5 in array

                fiveIndex=i;

        else if(arr[i]==8)

                eightIndex=i; // finding index of 8 in array

}
```

```
for(int i = fiveIndex ; i <= eightIndex ; i++)
{
        num2 += str[i];  //concatenating all values between 5 and 8 in the array.
        arr[i] = 0; //after concatenation we are assigning zero at that index because
        //we don't need that value any more.
}
for(int num:arr)
{
        num1 += num; //here we are adding all values. we have assigned zero in above loop
        //between fiveIndex and eightIndex in the arr.so it will add remaining outer values.
}
        System.out.println(num1+(Integer.parseInt(num2))); //print the sum!
}
}
```

Qno : 1 – Solution in Python

```
lst_input = input().split(",")
length = len(lst_input)
five_idx = lst_input.index('5')
eight_idx = lst_input.index('8')
concate = lst_input[five_idx:eight_idx+1]
sum_string = lst_input[0:five_idx] + lst_input[eight_idx+1:length]
sum_string = list(map(int,sum_string))
concate = ''.join(concate)
sum_remain = sum(sum_string)
print(int(concate)+sum_remain)
```

QUESTION 2:

A string which is a mixture of letters ,numbers and special characters from which produce the largest even number from the available digit after removing the duplicates digits.

If an even number did not produce then return -1.

Ex : **infosys@337**

O/p : **-1**

Hello#81@21349

O/p : **984312**

Qno : 2 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;
   public class Question2 {
      static boolean isDigit (char digit)
      {
         if(digit >= '0' && digit <= '9')
             return true;
         return false;
      }
      public static void main(String[] args)throws Exception {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
         String input=br.readLine();
         StringBuilder sb = new StringBuilder();
         HashSet<Integer> hs = new HashSet<>();
         for(int i = 0 ; i < input.length() ; i++)
         {
          if(isDigit(input.charAt(i)))
            {
               String dig = String.valueOf(input.charAt(i));
               hs.add( Integer.parseInt(dig) );
            }
         }
         ArrayList<Integer> digitList = new ArrayList<>(hs);
```

```
        Collections.sort(digitList, Collections.reverseOrder());
        int len = digitList.size();
        boolean evenDigitFound = false;
        for(int i = len-1 ; i >=0 ; i--)
        {
           int digit = digitList.get(i);
           if(digit % 2 == 0)
           {
              digitList.remove(i);
              digitList.add(digit);//adding evendigit at end
              evenDigitFound = true;
              break;
           }
        }
        if(evenDigitFound == false)
        {
           System.out.print(-1);
        }
        else
        {
        for(int dig : digitList)
           System.out.print(dig); // printing largest evendigits
        }
}
}
```

Qno : 2 – Solution in Python

```
import re
string=input()
digits=list(set(re.findall("\d",string)))
digits.sort(reverse=True)#sorting asc. order
num=int(''.join(digits)) #joining all the digits into one
if(num%2==0):
    print(num)
else:
    length=len(digits)
    for i in range(length-1,0,-1):
        if(int(digits[i])%2==0):
            d=digits[i]
            digits.remove(d)
            digits.append(d)
            even_num=int(''.join(digits))
            print(even_num)
            break
    else:
        print("-1")
```

QUESTION 3:

Read ‘m’ and ‘n’.

Take m*n matrix.

If any number is consecutive 3 times either in row ,column ,diagonals print that number , if there are multiple numbers then print min of those numbers.

Ex: m=6 ,n=7,take 6*7 matrix

2 3 4 5 6 2 4

2 3 4 7 6 7 6

2 3 5 5 5 5 2

2 3 1 1 2 1 3

1 1 1 1 9 0 3

2 3 1 1 5 1 2

O/p : 1

Qno : 3 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Collections;
```

```
public class Question3 {

    public static void main(String[] args)throws Exception {

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        String str[]=br.readLine().split(" ");//first line will row and column value separated with space

        int row=Integer.parseInt(str[0]);

        int column=Integer.parseInt(str[1]);

        int matrix[][]=new int[row][column];

        for(int i=0;i<row;i++)//every iteration we will take a row as an input.

        {

            String rowData[]=br.readLine().split(" ");

            for(int j=0;j<column;j++)

            {

                matrix[i][j]=Integer.parseInt(rowData[j]);

            }

        }


        ArrayList<Integer>list=new ArrayList<>();

        for (int r=0;r<row;r++)

            {

             for(int c=0;c<column;c++){

                 if(c < column-3) //condition for cosecutive numbers in all rows in matrix

                    {

                        if(matrix[r][c]==matrix[r][c+1] && matrix[r][c+2]==matrix[r][c+3] && matrix[r][c+1]==matrix[r][c+2])

                            list.add(matrix[r][c]);

                       }

                if(r < row-3) //   #condition for cosecutive numbers in all columns in matrix

                    {

                        if(matrix[r][c]==matrix[r+1][c] && matrix[r+2][c]==matrix[r+3][c] && matrix[r+1][c]==matrix[r+2][c])

                            list.add(matrix[r][c]);

                     }
```

```
            if(c < column-3 && r >= 3) //#cosecutive numbers in all left to right diagonals in matrix
              {
                 if(matrix[r][c]==matrix[r-1][c+1] && matrix[r-2][c+2]==matrix[r-3][c+3] && matrix[r-1][c+1]==matrix[r-2][c+2])
                     list.add(matrix[r][c]);
               }
            if(c >= 3 && r >= 3)  // #cosecutive numbers in all left to right diagonals in matrix
              {
                 if(matrix[r][c]==matrix[r-1][c-1] && matrix[r-2][c-2]==matrix[r-3][c-3] && matrix[r-1][c-1]==matrix[r-2][c-2])
                    list.add(matrix[r][c]);
               }
       }
     }
     if(list.size()==0)
        System.out.println("-1");
     else
        System.out.println(Collections.min(list));
}
}
```

Qno : 3 – Solution in Python

```
row,column=map(int,input().split(" "))
list_digit=[]
matrix=[]
for r in range(row):
    row_numbers=list(map(int,input().split()))
    matrix.append(row_numbers)
for r in range(0,row):
    for c in range(0,column):
        if(c < column-3): #condition for cosecutive numbers in all rows in matrix
                if(matrix[r][c]==matrix[r][c+1]==matrix[r][c+2]==matrix[r][c+3]):
                        list_digit.append(matrix[r][c])
```

```
        if(r < row-3):    #condition for cosecutive numbers in all columns in matrix
                if(matrix[r][c]==matrix[r+1][c]==matrix[r+2][c]==matrix[r+3][c]):
                        list_digit.append(matrix[r][c])

        if(c < column-3 and r >= 3): #cosecutive numbers in all left to right diagonals in matrix
                if(matrix[r][c]==matrix[r-1][c+1]==matrix[r-2][c+2]==matrix[r-3][c+3]):
                        list_digit.append(matrix[r][c])

        if(c >= 3 and r >= 3):   #cosecutive numbers in all left to right diagonals in matrix
                if(matrix[r][c]==matrix[r-1][c-1]==matrix[r-2][c-2]==matrix[r-3][c-3]):
                        list_digit.append(matrix[r][c])

if(len(list_digit)==0):
    print("-1")
else:
    print(min(list_digit))
```

QUESTION 4:

Take input a number 'N' and an array as given below.
Input:-
N = **2**
Array =**1,2,3,3,4,4**
O/p : 2
Find the least number of unique elements after deleting N numbers of elements from the array.
In the above example , after deleting N=2 elements from the array.
In above 1,2 will be deleted.
So 3,3,4,4 will be remaining so,
2 unique elements are in the array i.e 3 and 4.

Qno : 4 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.HashSet;

public class Question4 {
   public static void main(String[] args)throws Exception {
```

```
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine());
        String strArr[] = br.readLine().split(",");
        int len = strArr.length;
        HashSet <Integer> hs = new HashSet<>();
        for(int i = 0 ; i < len ; i++)
        {
           hs.add( Integer.parseInt(strArr[i] ));
        }
        System.out.println(hs.size()-n);
    }
}
```

Qno : 4 – Solution in Python

```
rem=int(input())
list_num=list(map(int,input().split(",")))
distinct = set(list_num)
print(len(distinct)-rem)
```

QUESTION 5:

String Rotation

Input :- **rhdt:246,ghftd:1246**

Output:- **trdt ftdgh**

Explanation :here every string is associated with the number separated with ':' if sum of squares of corresponding digits is even then rotate the string left by 1 position or if sum of squares of digits is odd then rotate the string right by 2 position.

*2*2+4*4+6*6=56* which is even so rotate left by 1 position **rhdt --->trhd.**

*1*1+2*2+4*4+6*6=57* which is odd then rotate right by 2 position **ghftd---> ftdgh**

Qno : 5 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Question5 {

   public static void main(String[] args)throws Exception {

      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

      String str[]=br.readLine().split(",");

      for(String obj:str)
```

```
    {
      String ar[]=obj.split(":");
      String s=ar[0];
      int number=Integer.parseInt(ar[1]);
      int digSqSum=0;
      while(number!=0)//adding every sq of digits
      {
        int digit=number%10;
        digSqSum+=(digit*digit);
        number/=10;
      }
      int lenOfString=s.length();
      if(digSqSum%2==0)
        System.out.print(s.charAt(lenOfString-1)+s.substring(0,lenOfString-1)+" ");
        //first printing last char and then concatenating all char except last
      else
        System.out.println(s.substring(2,lenOfString)+s.substring(0,2)+" ");
      //first printing substring from 3rd position and concatenatiin first two char in it.
    }
  }
}
```

Qno : 5 – Solution in Python

```
dictionary=input().split(",")
for obj in dictionary:
    str_obj=obj.split(":")
    string=str_obj[0]
    length=len(string)
    num=str_obj[1]
    sum=0
    for digit in num:
```

```
        sum += (int(digit)**2)
    if(sum%2==0):
        last_char = string[length-1]
        print(last_char+string[0:length-1],end=' ')
    else:
        first_2char = string[0:2]
        print(string[2:length]+first_2char,end=' ')
```

QUESTION 6:

Given an array of n elements. Create a square matrix and Print all square sub matrix whose sum is highest.

Input :

0,3,6,12,3,6,-12,3,-3

---- Possible square Matrix from this input-

0 3 6
12 3 6
-12 3 -3

Highest Possible Sum is 18 in this above matrix.

Now Print all square submatrix whose sum is highest i.e **18.**

Output:

0 3
12 3

3 6
3 6

0 3 6
12 3 6
-12 3 -3

Explanation :- There are 3 possible square submatrix whose sum are **18**.

Qno : 6 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Question6 {
   static int maxSum(int matrix[][] , int rc , int r , int c , int d)
   {
      int sum = 0;
      int r2 = r;
      int c2 = c;
      for(int i = 0 ; i < d+1  ; i++)
      {
         for(int j = 0 ; j < d+1 ; j++)
```

```
      {
        sum+=matrix[r2][c2];
         c2++;
      }
      r2++;
      c2 = c;
   }
   return sum;
}
static void printMatrix(int matrix[][] , int rc , int r , int c , int d, int maxSum)
{
   //printing that matrix whose sum is equal to maxSum.
   int matrixSum = 0;
   int r2 = r;
   int c2 = c;
   StringBuilder sb = new StringBuilder();
   for(int i = 0 ; i < d+1 ; i++)
   {
      for(int j = 0 ; j < d+1 ; j++)
      {
        matrixSum+=matrix[r2][c2];
        sb.append(matrix[r2][c2]+" ");
        c2++;
      }
      r2++;
      c2 = c;
      sb.append("\n");

   }
   if(matrixSum == maxSum)
   {
      System.out.println(sb.toString()); // matrix printing
   }
}
public static void main(String[] args)throws Exception {
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   String rcValues[] = br.readLine().split(",");
   int len = rcValues.length;
   int rc = (int)Math.sqrt(len);
   int matrix [][] = new int[rc][rc];
   int idx = 0;
   for(int i = 0 ; i < rc ; i++)
   {
      for(int j = 0 ; j < rc ; j++)
      {
         matrix[i][j] = Integer.parseInt(rcValues[idx]);
         idx++;
```

```
        }
      }
      int maximumSum = -1000000;
      //finding max sum
      for(int i = 0 ; i < rc ; i++)
      {
        for(int j = 0 ; j < rc ; j++)
        {
          for(int sqmt = 1 ; (sqmt+i < rc && sqmt+j < rc) ; sqmt++)
          {
            int sum = maxSum(matrix, rc, i, j, sqmt); // finding max sum
            if(sum > maximumSum )
              maximumSum = sum;
          }
        }
      }
     //code for printing all submatrix whose sum is max
     for(int i = 0 ; i < rc ; i++)
      {
        for(int j = 0 ; j < rc ; j++)
        {
          for(int sqmt = 1 ; (sqmt+i < rc && sqmt+j < rc) ; sqmt++)
          {
            printMatrix(matrix , rc, i, j, sqmt, maximumSum);
          }
        }
      }
    }
}
```

Qno : 6 – Solution in Python

```
import math

def maxSum(matrix, rc, r, c, d):

    matrix_sum = 0

    r2 = r

    c2 = c

    for i in range(d+1):

      for j in range(d+1):

        matrix_sum += matrix[r2][c2]

        c2+=1

      r2+=1
```

```
      c2 = c
    return matrix_sum
def printMatrix(matrix, rc, r, c, d, max_sum):
    matrix_sum = 0
    r2 = r
    c2 = c
    sub_matrix = ""
    for i in range(d+1):
      for j in range(d+1):
        matrix_sum += matrix[r2][c2]
        sub_matrix += str(matrix[r2][c2])+" "
        c2+=1
      r2+=1
      c2 = c
      sub_matrix += "\n"
    if(matrix_sum == max_sum):
      print(sub_matrix)


lst_input = list(map(int,input().split(",")))
length = len(lst_input)
rc = int(math.sqrt(length)) #row and column value
matrix = []
for i in range(0,length,rc):
  matrix.append(lst_input[i:rc+i])
max_sum = -1000000
for i in range(rc):
  for j in range(rc):
    for sqmt in range(rc):
      if(sqmt+i < rc and sqmt+j < rc):
        matrix_sum = maxSum(matrix, rc, i, j, sqmt)
```

```
        if(matrix_sum > max_sum):
          max_sum = matrix_sum


for i in range(rc):
  for j in range(rc):
    for sqmt in range(rc):
      if(sqmt+i < rc and sqmt+j < rc):
        printMatrix(matrix, rc, i, j, sqmt, max_sum)
```

QUESTION 7:

Given input of array of string in format <emp name>: <emp number> separated by comas.
You have to generate password for
Input : **Robert:36787,Tina:68721,Jo:56389**
Output : **tiX**
Find max digit in the corresponding emp number which is less or equal to the length of emp name.And add that place char into the final string .If there is no any digit which satisfy the condition then add 'X' into the final string.
Explanation:
So first there is **Robert:36787**

Now in the emp number there is **6** which is less than or equal to the length of **'Robert'.** So we will take 6th character of **'Robert'**. So here at 6th position there is **'t'** so we will add **'t'** into the final output string.

Same we will do with other empname and empnumber.
If you will observer **Tina : 68721**
Now in the emp number there is **2** which is less than or equal to the length of **'Tina'.** So we will take **2nd** character of **'Tina'**. So here at **2nd** position there is **'i'** so we will add **'i'** into the final output string.
If you will observe **Jo:56389**
So in the emp number there is no any digit which is less than or equal to length of **'Jo'** so in this case we will add 'X' into the final String Output.

Qno : 7 - Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Question7{
```

```
public static void main(String[] args)throws Exception {
   BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
   String str[]=br.readLine().split(",");
   String password="";
   for(String obj:str)
   {
      String ar[]=obj.split(":");
      String empName=ar[0];
      String empNumber=ar[1];
      int nameLen=empName.length();
      int numberLen=empNumber.length();
      int num=-1;
      for(int i=0;i<numberLen;i++)
      {
         int digit=Integer.parseInt(String.valueOf(empNumber.charAt(i)));
         if(digit<=nameLen && digit > num)
            num=digit;
      }
      if(num==-1)
         password+="X";
      else
         password+=String.valueOf(empName.charAt(num-1));

   }
   System.out.println(password);
}}
```

Qno : 7 - Solution in Python

```
input_string=input()
list_input=[]
finalStr=''
list_input=input_string.split(',')
for i in list_input:
    temp=i.split(':')
    name=temp[0]
    number=temp[1]
    length=len(name)
    max=0
    for digit in number:
        if(int(digit)<=length ):
            if(max<int(digit)):
                max=int(digit)
    if(max==0):
        finalStr+='X'
    else:
       finalStr+=name[max-1]
print(finalStr)
```

QUESTION 8:

A non empty string containing only brackets (,),{.},[,] it returns integer output based on following.

- If Input string is properly nested (means balanced) then return 0.
- If Input string is not properly nested ,return position of bracket of input string from where it is not balanced -position start from 1.

Input : **{([])}[]** output : 0 (Because every opening has a corresponding closing.)

Input : **([)()]** output :3

Input :**[[()]** output: n+1 for last element i.e 5+1 =6 (There is no closing of first bracket at the last, so we will print last position. That is 5+1=6).

Qno : 8 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.Stack;

public class Question8 {

   public static void main(String[] args)throws Exception {

      BufferedReader brr = new BufferedReader(new InputStreamReader(System.in));
```

```
String bracket = brr.readLine();
Stack<Character> stack = new Stack<>();
int len = bracket.length();
boolean flag = true;
for(int i = 0 ; i < len ; i++)
{
   char br = bracket.charAt(i);
   if(br == '(' || br == '{' || br == '[')
   {
      stack.push(br);
   }
   else if(stack.size() == 0)
   {
      System.out.println(i+1);
      flag = false;
      break;
   }
   else if(br == ')')
   {
      char pop = stack.pop();
      if(pop != '(')
      {
         System.out.println(i+1);
         flag = false;
         break;
      }
   }
```

```
    else if(br == '}')
    {
        char pop = stack.pop();
        if(pop != '{')
        {
            System.out.println(i+1);
            flag = false;
            break;
        }
    }
    else if(br == ']')
    {
        char pop = stack.pop();
        if(pop != '[')
        {
            System.out.println(i+1);
            flag = false;
            break;
        }
    }
}
if(stack.size() != 0 && flag == true)
{
    System.out.println(len+1);//brackets are unbalanced at the end
    flag = false;
}
if(flag)
```

```
        System.out.println("0");//means every opening has a corresponding closing

    }

}
```

Qno : 8 – Solution in Python

```python
brackets = input()
stack = []
length = len(brackets)
flag = True
for i in range(length):
  br = brackets[i]
  if(br == '(' or br == '{' or br == '['):
    stack.append(br)
  elif(len(stack) == 0):
      print(i+1)
      flag = False
      break
  elif( br == ')'):
    pop_val = stack.pop()
    if(pop_val != '(' ):
      print(i+1)
      flag = False
      break
  elif( br == '}'):
    pop_val = stack.pop()
    if(pop_val != '{' ):
      print(i+1)
      flag = False
      break
  elif( br == ']'):
    pop_val = stack.pop()
    if(pop_val != '[' ):
      flag = False
      print(i+1)
      break
if(len(stack)!=0 and flag == True):
  print(length+1)#brackets are unbalanced at the end
  flag = False
if(flag):#means every opening has a corresponding closing
  print("0")
```

QUESTION 9:

A non empty string containing only alphabets . print the longest prefix from input string which is same as suffix.
Prefix and Suffix should not be overlapped.
Print -1 if no prefix exist which is also the suffix without overlap.
Do case sensitive comparison.
Position start from 1.

Input : **xxAbcxxAbcxx**
o/p : **xx** ('xx' in the prefix and '**xx'** in the suffix and this is longest one in the input string so output will be 'xx').
Input : Racecar

o/p: **-1** (There is no prefix which is in also suffix so output will be -1).

Qno : 9 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Question9 {

   public static void main(String[] args)throws Exception {

      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

      String str=br.readLine();

      int length=str.length();

      int half=length/2;


      for(int i=half ; i>=0 ; i--)

      {

         String prefix=str.substring(0,i);

         String suffix=str.substring(length-i,length);

         if(prefix.equals("")||suffix.equals(""))

         {

            System.out.println("-1");

            break;
```

```
        }
        if(prefix.equals(suffix))
        {
           System.out.println(prefix);
           break;
        }
     }
  }
}
```

Qno : 9 – Solution in Python

```
string=input()
length=len(string)
half=length//2
for i in range(half,0,-1):
    prefix=string[0:i]
    suffix=string[length-i:length]
    if(prefix==suffix):
        print(prefix)
        break
else:
    print("-1")
```

QUESTION 10:

Number of odd sub arrays.

Find the number of distinct subarrays in an array of integers such that the sum of the subarray is an odd integer , two subarrays are considered different if they either start or end at different index.

Input:

1

3

1 2 3

Output:

4

Explanation : Total subarrays are [1], [1, 2], [1, 2, 3], [2], [2, 3], [3]

In this there is four subarrays which sum is odd i.e: [1],[1,2] ,[2,3],[3].

Qno : 10 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.HashSet;
public class Question10 {
   public static void main(String[] args)throws Exception {
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      int testcase=Integer.parseInt(br.readLine());
      while(testcase--!=0)
      {
      int n=Integer.parseInt(br.readLine());
      String str[]=br.readLine().split(" ");
      HashSet <String> hs = new HashSet<>();
      int arr[]=new int[n];
      for(int i=0;i<n;i++)
         arr[i]=Integer.parseInt(str[i]);
      for(int i=0;i<n;i++)
      {
         int sum = arr[i];
         if(sum % 2 != 0)
             hs.add(arr[i]+" ");
         StringBuilder sb = new StringBuilder();
         sb.append(arr[i]);
         for(int j=i+1;j<n;j++)
         {
            sum+=arr[j];
            sb.append(arr[j]);
            if(sum%2!=0)
               hs.add(sb.toString());
          }
      }
      System.out.println(hs.size());//that will be the no of unique subarray
      }
   }
}
```

Qno : 10 – Solution in Python

```
from itertools import combinations
t = int(input())
distinct_subarray = set()
while(t != 0):
  n = int(input())
```

```
  lst_input = list(map(int,input().split()))
  for i in range(n):
    sum = lst_input[i]
    string =''
    if(sum % 2 != 0):
      distinct_subarray.add(str(sum))
    string+=str(sum)
    for j in range(i+1,n):
      sum+=lst_input[j]
      string+=str(lst_input[j])
      if(sum % 2 != 0):
        distinct_subarray.add(string)
  t-=1
print(len(distinct_subarray))
```

QUESTION 11:

Find the all possible 2*2 matrix , and

Follow rule that each element of 2*2 matrix should be divisible by sum of its digits.

Input:-

4 3 (row value is 4 and column value is 3).

42 54 2

30 24 27

180 190 40

11 121 13

O/p:

42 54

30 24

30 24

180 190

24 27

190 40

Qno : 11 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.ArrayList;

public class Question11 {

    static int Sum(int num)

    {

       int temp=num;
```

```
    int digSum=0;
    while(temp!=0)
    {
        digSum+=temp%10;
        temp/=10;
    }
    if(num%digSum==0)
        return 0;
    else
        return 1;
}
public static void main(String[] args)throws Exception {
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 String str[]=br.readLine().split(" ");//first line will row and column value separated with space
 int row=Integer.parseInt(str[0]);
 int column=Integer.parseInt(str[1]);
 int matrix[][]=new int[row][column];
 for(int i=0;i<row;i++)//every iteration we will take a row as an input.
 {
    String rowData[]=br.readLine().split(" ");
    for(int j=0;j<column;j++)
    {
        matrix[i][j]=Integer.parseInt(rowData[j]);
    }
 }
    System.out.println("");
 for (int r=0;r<row-1;r++)
```

```
        {
         for(int c=0;c<column-1;c++){
           int num1=Sum(matrix[r][c]);
           int num2=Sum(matrix[r][c+1]);
           int num3=Sum(matrix[r+1][c]);
           int num4=Sum(matrix[r+1][c+1]);
           if(num1==num2 && num2==num3 && num3==num4 ){
              System.out.println(matrix[r][c]+" "+matrix[r][c+1]);
              System.out.println(matrix[r+1][c]+" "+matrix[r+1][c+1]);
           }
        }
   }
}
}
```

Qno : 11 – Solution in Python

```
def is_divisible(n):
    s = sum(list(map(int,str(n))))
    if n % s==0:
        return True
    else:
        return False

row ,col = list(map(int,input().split()))
matrix = []
for i in range(row):
    matrix.append(list(map(int,input().split())))
for r in range(row-1):#3
    for c in range(col-1):#2
        if(is_divisible(matrix[r][c]) and is_divisible(matrix[r][c+1]) and is_divisible(mat
rix[r+1][c]) and is_divisible(matrix[r+1][c+1])):
            print(matrix[r][c],matrix[r][c+1])
            print(matrix[r+1][c],matrix[r+1][c+1])
```

QUESTION 12:

Max Subarray

An array is given suppose arr =3,5,8,2,19,12,7,13

One have to find the largest subarray that the element satisfy the following condition

x[i]=x[i-1]+x[i-2]

If more than one subarray found then largest one has to be print.And if two subarrays has same number of elements then we will print that subarray which will start first.

Here Expected output is [3,5,8,13].

Explanation :- 3rd element is 8 because sum of previous 2 elements (3 and 5), is 8 and 8 is in the input array.

Same like that 4th element will be the 13. Bacause sum of previous 2 elements (8 and 5) ,is 13 and 13 is in the input array.

Qno : 12 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Collections;

public class Question12 {

   public static void main(String[] args)throws Exception {

      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

      String strArr[]=br.readLine().split(",");

      ArrayList<Integer> list=new ArrayList<>();

      for(String s:strArr)

         list.add(Integer.parseInt(s));

      ArrayList<Integer> longestSeries=new ArrayList<>();

      int n=list.size();

      for(int i=0;i<n-1;i++)

      {

         int first=list.get(i);

         int second=list.get(i+1);

         ArrayList<Integer> fabList=new ArrayList<>();

         fabList.add(first);

         fabList.add(second);

         for (int j=i+2;j<n;j++){
```

```
        int ele=list.get(j);
        if(first+second==ele){
          fabList.add(ele);
          first=second;
          second=ele;
        }
        }
        if(longestSeries.size()<fabList.size())
          longestSeries=fabList;
     }
if(longestSeries.size()>2)
{
   for(int ele:longestSeries)
      System.out.print(ele+" ");
}
else
    System.out.println("-1");
   }
}
```

Qno : 12 – Solution in Python

```
list_number=list(map(int,input().split(",")))
length=len(list_number)
list_total=[]
for i in range(0,length):
    for x in range(i+1,length):
        first=list_number[i]
        second=list_number[x]
        fab_list=[]
        fab_list.append(first)
        fab_list.append(second)
        for y in range(x+1,length):
```

```
            if(first+second==list_number[y]):
               fab_list.append(list_number[y])
               first=second
               second=list_number[y]
        if(len(list_total)<len(fab_list)):
             list_total=fab_list
if(len(list_total)>2):
     for ele in list_total:
          print(ele,end =' ')
else:
     print("-1")
```

QUESTION 13:

A string is given we have to find the longest substring which is unique (that has no repetition) and min size should be 3.
If more than one sub string is found with max length then we have to print one which appeared first in the string.
If no substring is present which matches the condition then we have to print -1;
Note : output may be case insensitive (means A & a will be treated as different character, same with other alphabets also.)
Ex :input : **A@bcd1abx**
Output : **A@bcd1a**

Qno : 13 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.HashSet;
/**
 *
 * @author deepukumar
 */
public class Question13 {
   public static void main(String[] args)throws Exception {
      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
      String input = br.readLine();
      int length = input.length();
      int lenOfUnique = 0;
```

```
    ArrayList<String> unique = new ArrayList<>();
    for (int i = 0 ; i < length ; i++)
    {
       ArrayList<String> subString = new ArrayList<>();
       subString.add(input.charAt(i)+"");
       for (int j = i+1 ; j < length ; j++)
       {
          if(subString.indexOf(input.charAt(j)+"") == -1)
          {
             subString.add(input.charAt(j)+"");
          }
          else
          {
          int subLen = subString.size();
          if(subLen >= 3)
          {
             if(unique.size() < subLen)
             {
                unique = subString;
             }
          }
          break;
          }
       }
    }
    if(unique.size()<3)
       System.out.println("-1");
    else
    {
    for(String ch : unique)
       System.out.print(ch);
    }

}}
```

Qno : 13 – Solution in Python

```
string=input()
length=len(string)
unique=''
for i in range(length):
    substring=string[i]
    for j in range(i+1,length):
        substring+=string[j]
        sub_len=len(substring)
        if(sub_len>=3 and len(set(substring))==sub_len):
            if(len(unique)<sub_len):
                unique=substring
if(len(unique)==0):
    print("-1")
else:
    print(unique)
```

QUESTION 14:

Write a function called nearest_palindrome ()

Which can accepts a number and return the nearest greater palindrome number .

Input : 12300 --> Output :- 12321

Input : 12331 --> Output :- 12421

Qno : 14 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Question14 {

   public static void main(String[] args)throws Exception {

      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

      int num = Integer.parseInt(br.readLine());
```

```
    num++;
    while(true)
    {
      StringBuffer sb = new StringBuffer(String.valueOf(num));
      String str = sb.toString();
      String strRev = sb.reverse().toString();
      if(str.equals(strRev))
      {
        System.out.println(str);
        break;
      }
      else
        num++;
    } }}
```

Qno : 14 – Solution in Python

```python
num = int(input())
num+=1
while(True):
  if(str(num) == str(num)[::-1]):
    print(num)
    break
  else:
    num+=1
```

QUESTION 15:

A string is a combination of alphabets , numbers and special characters. You have to reverse that string by keeping the special characters in the same position.

Input 1:

b@rd

output 1:

d@rb

Input 2:
ajh@#jsk75
Output 2:
57k@#sjhja

Qno : 15 - Solution in Java

```
import java.util.Scanner;

public class Question15 {
   static boolean isAlphaNumeric(char ch )
   {
      if(ch >= 'a' && ch <= 'z')
         return true;
      else if(ch >= 'A' && ch <= 'Z')
         return true;
      else if(ch >= '0' && ch <= '9')
         return true;

      return false;
   }

   public static void main(String[] args)throws Exception {
      Scanner kb = new Scanner(System.in);
      String input = kb.nextLine();
      int len = input.length();
      char reverse[] = new char[len];
      int idx = 0;
      StringBuilder sb = new StringBuilder();
      String rev="";
      for(int i = len-1 ; i >= 0 ; i--)//reverse traversing
      {
         if(isAlphaNumeric(input.charAt(i)))
         {
            sb.append(input.charAt(i));

         }
      }
      for(int i = 0 ; i < len ; i++)
      {
         if(!isAlphaNumeric(input.charAt(i)))
            sb.insert(i,input.charAt(i));
      }
      System.out.print(sb);
```

```
  }
}
```

Qno : 15 - Solution in Python

```
string=input()
string_list=[]
for i in string:
    if(i.isalnum()):
      string_list.append(i)
string_list.reverse()
length = len(string)
for i in range(length):
    if(string[i].isalnum() == False):
        string_list.insert(i,string[i])
print(''.join(string_list))
```

QUESTION 16:

OTP Generation

Input Format: 134567

Output Format:1925

Explanation:

Take the string of numbers and generate a four

digit OTP such that.

1.If the number is odd square it.

2.If the number is even ignore it.

So in the input number 1,3,5,7 are odd.

Square each digits we will get.192549 now print first four digits as output. i.e 1925.

If in case if it is not possible then print -1.

Qno : 16 – Solution in Java

```
import java.util.Scanner;
public class Question16{
   public static void main(String[] args) {
      Scanner kb=new Scanner(System.in);
      String input=kb.next();
      StringBuilder sb=new StringBuilder();
      int sq=0;
     int length = input.length();
      for(int i=0;i<length;i++){
        char ch=input.charAt(i);
        int no=Integer.parseInt(String.valueOf(ch));
         if( no % 2!=0){
           sq=no*no;
```

```
            sb.append(String.valueOf(sq));
          }
        }
        String str=sb.toString();
        if(str.length()<4){
           System.out.println("-1");
        }
        else{
        String output=str.substring(0,4);
        System.out.println(output);
        }
    }

}
```

Qno : 16 – Solution in Python

```
number=input()
length=len(number)
otp=''
for i in range(length):
    if(int(number[i]) % 2 != 0):
        otp+= str(int(number[i])**2)
if(len(otp)<4):
    print("-1")
else:
    print(otp[0:4])
```

QUESTION 17:

Input 1:- Asp5w8w@k7!l23mn69

Output:- 8527639

If number of special characters in the given string is **even** so we will print first even digit and next odd digit in the same series as they are present in the string as shown above (input 1).

If number of special characters in the given string is **odd** so we will print first odd digit and next even digit in the same series as they are present in the string as shown below(input 2).

Input 2:- #bn7856!@kh48522

Output:-785654822 (At the last there is no more odd digits so we will print remaining even digits as they are present in the input string).

Qno : 17 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;
public class Question17 {
```

```
static boolean isAlphabet(char ch)
{
   if(ch>='a' && ch<='z')
      return true;
   else if(ch>='A' && ch<='Z')
      return true;
   return false; // that means char ch is special char
}
static void printDigits(ArrayList<Integer> list1 , ArrayList<Integer>list2)
{
   int listLength1 = list1.size();
   int listLength2 = list2.size();
   int max = Math.max(listLength1, listLength2);
   int li = 0;
   int li2 = 0;
   for(int i = 0 ; i < max ; i++)
   {
      if(li != listLength1)
      {
         System.out.print(list1.get(i));
         li++;
      }
      if(li2 != listLength2)
      {
         System.out.print(list2.get(i));
         li2++;
      }
   }
}
public static void main(String[] args)throws Exception {
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   String input = br.readLine();
   ArrayList<Integer> oddList = new ArrayList<>();
   ArrayList<Integer> evenList = new ArrayList<>();
   int specialCharCount = 0;
   int len = input.length();
   for(int i = 0 ; i < len ; i++)
   {
      char ch = input.charAt(i);
      if(ch >= '0' && ch <= '9')
      {
         int dig = Integer.parseInt(String.valueOf(ch));
         if(dig % 2 == 0)
```

```
        {
          evenList.add(dig);
        }
        else
        {
          oddList.add(dig);
        }
      }
      else if(isAlphabet(ch) == false)
      {
        specialCharCount++;
      }
    }
    if(specialCharCount % 2 != 0)
      printDigits(evenList , oddList); // even odd will print alternate.
    else
      printDigits(oddList , evenList); // odd even will print alternate.
  }
}
```

Qno : 17 – Solution in Python

```
string=input()
even=[]
odd=[]
special_char=0
for ch in string:
    if(ch.isalnum() == False):
        special_char+=1
    elif(ch.isdigit()):
        if int(ch)%2==0:
            even.append(ch)
        else:
            odd.append(ch)

if(special_char%2==0):
      odd,even=even,odd
even_len=len(even)
odd_len=len(odd)
m=max(even_len,odd_len)
e=0
o=0
for i in range(m):
    if(e!=even_len):
        print(even[e],end='')
        e+=1
```

```
    if(o!=odd_len):
        print(odd[o],end='')
        o+=1
```

QUESTION 18:

Input:- 93012630
Output:- 2 6 12 30 930
We should take every possible substrings and we will verify each substring is a pronic number or not if pronic we will print that number.(Output Numbers should be unique).
Pronic Number: A number which is multiple of two consecutive integers is called Pronic Number.
6->2*3 it's a pronic
12->3*4 it's a pronic
Input: 12665042
Output:- 2 6 12 42 650

Qno : 18 – Solution in Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;
public class Question18 {
  static boolean isPronic(int number)
  {
    int sqrtNum = (int) Math.sqrt(number);
    int product = sqrtNum * (sqrtNum + 1);
    if (product == number)
      return true;
    else
      return false;
  }
  public static void main(String[] args)throws Exception {
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    String input = br.readLine();
    int len = input.length();
    HashSet<Integer> pronicList = new HashSet<>();
    for(int i = 0 ; i < len ; i++)
    {
      int number = Integer.parseInt(input.charAt(i)+"");
      if(isPronic(number) && number!=0)
      {
       pronicList.add(number);
      }
```

```
        for(int j = i+1 ; j < len ; j++)
        {
           int digit = Integer.parseInt(input.charAt(j)+"");
           number = number * 10 + digit;
           if(isPronic(number))
           {
              pronicList.add(number);
           }
        }
      }
      ArrayList<Integer> sorted = new ArrayList<>(pronicList); // converting set into ArrayList
      Collections.sort(sorted);
      for(int p : sorted)
      {
        System.out.print(p+" ");
      }
    }
}
```

Qno : 18 – Solution in Python

```
import math
def isPronic(num):
  sq = int(math.sqrt(num))
  product = sq * (sq+1)
  if(product == num):
    return True
  else:
    return False
string_num = input()
pronic_set = set()
length = len(string_num)
for i in range(length):
  num = string_num[i]
  if(isPronic(int(num)) and num != '0'):
    pronic_set.add( int(num))
  for j in range (i+1,length):
    num += string_num[j]
    if(isPronic( int(num) )):
      pronic_set.add(int(num))
for pr in sorted(pronic_set):
  print(pr,end=' ')
```

QUESTION 19:

An array of integers will be given and a sum is given.

Input:- **-1,1,0,0,2,-2**

Sum=0

Output: **3**

Output should be number of combination in given array which sum is equal to given sum and length should be 4 .So in our case these are three possible combination which sum is equal to given sum and their length is also 4.

(-1,1,2,-2)(0,0,1,-1)(0,0,-2,2)

So Output will be 3.

Qno : 19 – Solution in Java

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.ArrayList;

public class Question19 {

   public static void main(String[] args)throws Exception {

      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

      String strArr[] = br.readLine().split(",");

      int inputSum = Integer.parseInt(br.readLine());

      int n = strArr.length;

      int combination = 0;

      for (int i = 0; i < (1<<n); i++)

      {

         //finding subset

         ArrayList<Integer> subset = new ArrayList<>();

         for (int j = 0; j < n; j++)

         {

           // (1<<j) is a number with jth bit 1  // so when we 'and' them with the

            // subset number we get which numbers// are present in the subset and which are not
```

```
            if ((i & (1 << j)) > 0)
                subset.add(Integer.parseInt(strArr[j]));
        }
        int size = subset.size();
        if(size==0)
            continue;
        else if(size == 4)
        {
        int sum = subset.get(0) ;
        for(int x = 1 ; x < size ; x++)
            sum += subset.get(x);
        if(sum==inputSum)
            combination++;
        }
    }
    System.out.println(combination);
  }
}
```

Qno : 19 – Solution in Python

```
import itertools
input_set = list(map(int,input().split(",")))
s = int(input())
lst_combination=list(itertools.combinations(input_set,4))
count=0
for obj in lst_combination:
    obj_sum = sum(obj)
    if (obj_sum == s):
        count+=1
if(count==0):
    print("-1")
else:
    print(count)
```