EVEN SPLIT

```
t=int(input())
for _ in range(t):
    n=int(input())
    if n>2 and n&1==0:
        print('Yes')
    else:
        print('No')
```

Z PATTERN

```
for t in range(int(input())):
    n=int(input())
    print(f'Case #{t+1}:')
    print(n*"*")
    s=n-2
    for i in range(n-2):
        for j in range(n):
            if j==s:
                print("*",end="")
                s-=1
            else:
                print(" ",end="")
        print()
    print(n*"*")
```

NUMBER OF DIVISIORS

```
import math
t=int(input())
for _ in range(t):
    n=int(input())
    c=0
    for i in range(1,int(math.sqrt(n)+1)):
        if n%i==0:
            if n/i==i:
                c+=1
            else:
                c+=2
    print(c)
```

CONSECUTIVE BITS

```
for _ in range(int(input())):
    n=int(input())
    max=0
    curr=0
    while n:
        if n%2==1:
            curr+=1
        else:
            curr=0
        if max<curr:
            max=curr
        n=n//2
    print(max)
```

CEIL OR FLOOR OF A NUMBER

```
def floor(l,t):
   start=0
   end=len(l)-1
   mid=0
   while(start<=end):
      mid=start+(end-start)//2
      if(l[mid]==t):
         return l[mid]
      elif l[mid]<t:
         start=mid+1
      else:
         end=mid-1
   if start<len(l) and  l[start]>t:
      return l[start]
   else:
      return 2147483647

n=int(input())
l=list(map(int,input().split()))[:n]
q=int(input())
l.sort()
for i in range(q):
   t=int(input())
   print(floor(l,t))
```

RIGHT ANGLE PATTERN

```
import java.util.*;
public class Solution{
  public static void main(String[] args) {
      Scanner in=new Scanner(System.in);
      int t=in.nextInt();
      for(int j=1;j<=t;j++)
      {
          int n=in.nextInt();
          print(n,j);
      }
  }
  static void print(int n,int k)
  {
     System.out.println("Case #"+k+":");
     for(int i=0;i<n;i++)
     {
         for(int j=0;j<n-i-1;j++)
         System.out.print(" ");
         for(int m=0;m<=i;m++)
         System.out.print("*");
         System.out.println();
     }
  }
}
```

MATRIX ROTATION

```
        {
        int t=in.nextInt();
        int[][] arr=new int[t][t];
        for(int i=0;i<t;i++)
        for(int j=0;j<t;j++)
        arr[i][j]=in.nextInt();

         for(int i=0;i<t;i++)
         {
        for(int j=t-1;j>i;j--)
        {
          int temp=  arr[i][j];
           arr[i][j]=arr[j][i];
           arr[j][i]=temp;
        }
         }
         int n=0;
        if(t%2==0)
        n=t/2;
        else
        n=t/2+1;
         for(int i=0;i<t;i++)
         {
        for(int j=0;j<n;j++)
               {
           int temp=  arr[i][j];
           arr[i][j]=arr[i][t-j-1];
           arr[i][t-j-1]=temp;
               }
         }
    System.out.println("Test Case #"+k+":");
         for(int i=0;i<t;i++){
        for(int j=0;j<t;j++){
           System.out.print(arr[i][j]+" ");
        }
        System.out.println();
        }

    }
    }
  }
```

```
      int t=in.nextInt();
      for(int i=0;i<t;i++)
      {
      int n=in.nextInt();
//DIAGONAL TRAVERSAL
      int[][] arr=new int[n][n];
      for(int k=0;k<n;k++)
      for(int j=0;j<n;j++)
      arr[k][j]=in.nextInt();
      diagnal(arr);
       System.out.println();
  }
}
    public static void diagnal(int[][] arr)
    {
        int i=0,j=0;
        int n=arr.length;
        int sum=0;
        for(int k=n-1;k>=0;k--)
        {
            i=0;
            j=k;
            sum=0;
            while(j<n)
            {
                sum+=arr[i++][j++];
            }
              System.out.print(sum+" ");
        }
        for(int k=1;k<n;k++)
        {
            i=k;
            j=0;
            sum=0;
            while(i<n)
            {
                sum+=arr[i++][j++];
            }
            System.out.print(sum+" ");
        }
    }
```

```
int top=0,left=0;
        int right=n-1, bottom=n-1;
        while(top<=bottom && left<=right)
        {
            for(i=left;i<=right;i++)
         System.out.print(arr[left][i]+" ");
            for(i=top+1;i<=bottom;i++)
             System.out.print(arr[i][bottom]+" ");

            for(i=right-1;i>left;i--)
            {
            System.out.print(arr[right][i]+" ");
            }
            for(i=bottom;i>top;i--)
           System.out.print(arr[i][top]+" ");
           left++;
           top++;
           right--;
           bottom--;
        }
        System.out.println();
     }
  }
}
```

TRAILING ZEROS

```
t=int(input())
for i in range(t):
    n=int(input())
    k=0
    while n>0:
        k+=n//5
        n=n//5
    print(k)
```

LCM AND HCF or GCD

```
def gcd(a,b):
    if(b==0):
        return a
    return gcd(b,a%b)
t=int(input())
for _ in range(t):
    a,b=map(int,input().split())
    print((a*b)//gcd(a,b),gcd(a,b))
```

REVERSE BITS

```
import math
def rev(n):
    if n==0:
        return 0
    l=[]
    while n:
        l.append(n%2)
        n=n//2
    s=32-len(l)
    while s>0:
        l.append(0)
        s=s-1
    s=32
    ans=0
    for i in range(len(l)):
        if l[i]:
            ans+=pow(2,31-i)
    return ans
t=int(input())
for _ in range(t):
    print(rev(int(input())))
```

FLIP BITS

```
import math
def bin(n):
    s=""
    if n==0:
        return '0'
    else:
        while(n>0):
            s+=str(n%2)
            n=n>>1
        return s[::-1]
t=int(input())
for _ in range(t):
    a,b=map(int,input().split())
    a=bin(a)
    b=bin(b)
    c=0
    i=len(a)-1
    j=len(b)-1
    while i>=0 and j>=0:
        if a[i]!=b[j]:
            c+=1
        i-=1
        j-=1
    while i>=0:
        if a[i]=='1':
            c+=1
        i-=1
    while j>=0:
        if b[j]=='1':
            c+=1
        j-=1
    print(c)
```

SWAP BITS

```
import math
def binn(n):
    b=""
    while n:
        k=str(n%2)
        b=k+b
        n>>=1
    return b
def dec(s):
    j=len(s)-1
    ans=0
    for i in s:
        if i=='1':
            ans+=int(math.pow(2,j))
        j-=1
    return ans
for t in range(int(input())):
    n=int(input())
    s=binn(n)
    if len(s)&1:
            s='0'+s
    res=""
    i=0
    while i<len(s):
            res+=s[i+1]+s[i]
            i+=2
    print(dec(res))
```

TRIPLE DOUBLE

```
t=int(input())
for _ in range(t):
    n=int(input())
    l=list(map(int,input().split()))[:n]
    l.sort()

    i=0
    while i<n-2:
        if(l[i]==l[i+1] and l[i]==l[i+2]):
            i=i+3
            continue
        else:
            print(l[i])
            break
    else:
       print(l[-1])
```

REPEATED NUMBERS

```
t=int(input())
for _ in range(t):
    n=int(input())
    l=list(map(int,input().split()))[:n]
    l.sort()
    ans=[]
    for i in range(n-1):
        if l[i]==l[i+1]:
            ans.append(l[i])
    print(*ans)
```

XOR SUM OF PAIRS

```
for t in range(int(input())):
    n=int(input())
    l=list(map(int,input().split()))[:n]
    ans=0
    for i in l:
        ans=ans^(i<<1)
    print(ans)
```

BUBBLE SORT ADHOC

```
for _ in range(int(input())):
    n=int(input())
    ar=list(map(int,input().split()))[:n]
    c=0
    for i in range(n-1):
        for j in range(0,n-i-1):
            if ar[j+1]<ar[j]:
                c+=1
                ar[j+1],ar[j]=ar[j],ar[j+1]
    print(c)
```

SELECTION SORT ADHOC

```
for t in range(int(input())):
    n=int(input())
    l=list(map(int,input().split()))[:n]
    for i in range(n-1,0,-1):
        min=i
        j=i-1
        while j>=0:
            if l[min]<=l[j]:
                min=j
            j-=1
        l[i],l[min]=l[min],l[i]
        print(min,end=" ")
    print()
```

INSERTION SORT ADHOC

```
import java.util.*;
public class Main{
  public static void main(String[] args) {
       Scanner in=new Scanner(System.in);
     int t= in.nextInt();
     for(int k=0;k<t;k++){
       int n= in.nextInt();
       int[] s=new int[n];
        for(int i=0;i<n;i++)
            s[i]=in.nextInt();
            insertionSort(s);
     }
  }
  static void insertionSort(int[] nums)
  {
      int n=nums.length;
      int i=1,j=0;
      for(i=1;i<n;i++)
      {
         j=i-1;
         int temp=nums[i];
         while(j>=0 && nums[j]>temp)
         {
            nums[j+1]=nums[j];
            j--;
         }
         nums[j+1]=temp;
         System.out.print(j+1+" ");
      }
      System.out.println();
  }
}
```

SUM OF PAIRS

```
    HashMap<Integer,Integer> hmap=new HashMap<Integer,Integer>();
    for(int i=0;i<n;i++)
    {
      if(!hmap.containsKey(s[i]))
      {
        hmap.put(s[i],1);
      }
      else
      {
        int freq=hmap.get(s[i]);
        hmap.put(s[i],++freq);
      }
    }
    boolean check=true;
    for(int i=0;i<n;i++)
    {
      int temp=p-s[i];
      if(hmap.containsKey(temp))
      {
      if(!(s[i]==temp))
        {
          System.out.println("True");
          check=false;
          break;
        }
        if(s[i]==temp && hmap.get(s[i])>1)
        {
          System.out.println("True");
          check=false;
          break;
        }
      }
    }
    if(check)
     System.out.println("False");
    }
}
}
```

PAIR DIFFERENCE

```
    HashMap<Integer,Integer> hmap=new HashMap<Integer,Integer>();
    for(int i=0;i<n;i++)
    {
       if(!hmap.containsKey(s[i]))
       {
          hmap.put(s[i],1);
       }
       else
       {
          int freq=hmap.get(s[i]);
          hmap.put(s[i],++freq);
       }
    }
    boolean check=true;
    for(int i=0;i<n;i++)
    {
       int temp=p+s[i];
       if(hmap.containsKey(temp))
       {
       if(!(s[i]==temp))
          {
             System.out.println("true");
             check=false;
             break;
          }
          if(s[i]==temp && hmap.get(s[i])>1)
          {
             System.out.println("true");
             check=false;
             break;
          }
       }
    }
    if(check)
     System.out.println("false");
    }
}
}
```

FLOOR

```
def floor(l,t):
    start=0
    end=len(l)-1
    mid=0
    while(start<=end):
        mid=start+(end-start)//2
        if(l[mid]==t):
            return l[mid]
        elif l[mid]<t:
            start=mid+1
        else:
            end=mid-1
    if l[end]<t:
        return l[end]
    else:
        return -2147483648

n=int(input())
l=list(map(int,input().split()))[:n]
q=int(input())
l.sort()
for i in range(q):
    t=int(input())
    print(floor(l,t))
```

FIND FREQUENCY

```
n=int(input())
l=list(map(int,input().split()))
q=int(input())
d={}
for i in l:
    if d.get(i) is None:
        d[i]=1
    else:
        d[i]=d.get(i)+1
for i in range(q):
    k=int(input())
    if d.get(k) is None:
        print(0)
    else:
        print(d.get(k))
```

FIRST REPEATING CHARACTER

```
for t in range(int(input())):
    s=input()
    d={}
    t=True
    for i in s:
        if d.get(i) is None:
            d[i]=1
        else:
            d[i]+=1
    for i in s:
        if i in d and d[i]>=2:
            print(i)
            t=False
            break
    if t:
        print('.')
```

SUM OF SUBARRAYS

```
n=int(input())
l=list(map(int,input().split()))[:n]
cSum=[]
sum=0
for i in range(len(l)):
    sum=sum+l[i]
    cSum.append(sum)
for _ in range(int(input())):
    a,b=map(int,input().split())
    if a==0:
        print(cSum[b])
    else:
        print(cSum[b]-cSum[a-1])
```

SUBSTRING MATCHING

```
for _ in range(int(input())):
    a=input()
    b=input()
    q=int(input())
    for F in range(q):
        i,j,k,l=map(int,input().split())
        s1=a[i:(j+1)]
        #print(s1)
        b1=b[k:(l+1)]
        #print(b1)
        if s1 in b1:
            print("Yes")
        else:
            print("No")
```

ROTATION OF ARRAY

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        int t=in.nextInt();
        while(t-- > 0)
        {
            int n=in.nextInt();
            int k=in.nextInt();
            int[] nums=new int[n];
            int i=0;
            for(i=0;i<n;i++)
            nums[i]=in.nextInt();

            k=k%n;
            for(i=n-k;i<n;i++)
                System.out.print(nums[i]+" ");
            for(i=0;i<n-k;i++)
                System.out.print(nums[i]+" ");
            System.out.println();
        }
    }
}
```

AGGRESSIVECOWS

```
static int aggressiveCows(int[] arr,int c)
  {
int n=arr.length;
      Arrays.sort(arr);
      int low=arr[0],high=arr[n-1]-arr[0],mid;
      int ans=0;
      while(low<=high)
      {
         mid=low+(high-low)/2;
         if(placeCow(arr,c,mid))
         {
            ans=mid;
            low=mid+1;
         }
         else
             high=mid-1;
      }
      return ans;
  }
  static boolean placeCow(int[] arr,int cows,int distance)
  {
      int k=arr[0],count=1;
      for(int i=1;i<arr.length;i++)
      {
         if((arr[i]-k)>=distance)
         {
         count++;
         k=arr[i];
         }
         if(count==cows)
         return true;
      }
      return false;
  }
```

MAX SUB ARRAY KADANES

```
import java.util.*;
public class Main{
  public static void main(String[] args) {
      Scanner in=new Scanner(System.in);
    int t= in.nextInt();
    for(int k=0;k<t;k++){
      int n= in.nextInt();
      int[] s=new int[n];
       for(int i=0;i<n;i++)
          s[i]=in.nextInt();
          maxSubArray(s);
    }
  }
  static void maxSubArray(int[] nums)
  {
     int n=nums.length;
     int max=Integer.MIN_VALUE;
     int curr=0;
     int s=0,e=0;
     for(int i=0;i<n;i++)
     {
        curr+=nums[i];
        //System.out.println(curr);
        if(curr<0)
        {
           s=i;
           curr=0;
        }
        if(curr>max)
        {
           max=curr;
           e=i;
        }
     }
     System.out.println(max+" "+s+" "+e);
  }
}
```

PRIME FACTORS

```
static void printPrimeFactors(int n)
  {
     for(int i=2;i*i<=n;i++)
     {
        while(n%i==0)
        {
           System.out.print(i+" ");
           n=n/i;
        }
     }
     if(n>1)
     System.out.print(n+" ");
  }
```

SIEVE OF ERAS...

```
  static boolean[] sieve(int n)
  {
     boolean[] primes=new boolean[n+1];
     primes[0]=true;
     primes[1]=true;
     int i=0,j=0;
     for(i=2;i*i<=n;i++)
     {
        if(!primes[i])
        {
        for(j=i*i;j<=n;j+=i)
            primes[j]=true;
        }
     }
     return primes;
  }
```

https://github.com/SheetanshKumar/smart-interviews-problems/tree/master

https://github.com/jpallavi23/Smart-Interviews/tree/master/Filtering_Contest

https://github.com/jpallavi23/Smart-Interviews/tree/master/07_SI_Primary
-Hackerrank

https://github.com/adisayhi27/Hackerrank-SI
/tree/master