Unit5

https://oj.masaischool.com/contest/3524

Cipher String

Ended

Description

You are given a string of size N. You have to convert the string into its cipher form.

For example, the cipher form of a string "aabccd", will be "a2b1c2d1". The new generated string contains the characters, and the count of their occurrences in a consecutive manner.

Note: The string contains only lower-case characters.

```
function check(N,str)
{
   let count=0;
   bag=""
   for(i=0;i<N;i++)
   {
      if(str[i]==str[i+1])
      {
         count++
      }
      else{
         bag=bag+str[i]+(count+1)
         count=0
      }
   }
   console.log(bag)
}
function runProgram(input)
{
```

```
  input=input.trim().split('\n');
  var tc=+input[0];
  var line=1;
  for(let i=0;i<tc;i++)
  {
     N=+input[line]
     line++
     str=input[line]
     line++
    check(N,str)
   // console.log(N,str)
  }
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
```

```
});
}
```

Decipher String

Ended

Description

You are given a ciphered string, you have to decipher the string.

For example, if the given string is "a2b1c2", then the deciphered string will be "aabcc ".

Note: The string contains only lower-case letters and numbers.

```
function  check(N,arr)
{
   arr=arr.split("")
   let res=""
   for(let i=0;i<N;i++)
   {
      if(i%2===0)
      {
         res+=arr[i]
      }
      else if(i%2!==0)
      {
         let N=+arr[i]
         for(let j=1;j<N;j++)
         {
            res+=arr[i-1]
         }
      }
```

```
    }
    console.log(res)
}
function runProgram(input)
{
  input=input.trim().split('\n');
  var tc=+(input[0]);
  var line=1;
  for(i=0;i<tc;i++)
  {
     var N=+input[line]
     line++
     var arr=input[line]
     line++
    check(N,arr)
   // console.log(N,arr)
  }
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
```

```
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

String Cut

Ended

Description

You are given a string S. Cut it into 2 equal halves and reverse it.

So, suppose if you have a string "abcxyz" then after performing the above mentioned operation it becomes "cbazyx"

If you have a string "abcdxyz", then after performing the above mentioned operation i t becomes "cbadzyx"

```
function check(str)
{
    let bag=""
    if(str.length%2===0)
    {
        n=str.length/2
        for(let i=n-1;i>=0;i--)
        {
            bag+=str[i]
        }
        for(let j=str.length-1;j>=n;j--)
        {
            bag+=str[j]
```

```
        }
    }
    else
    {
        n=Math.floor(str.length/2)
        for(let i=n-1;i>=0;i--)
        {
            bag+=str[i]
        }
        bag+=str[n]
        for(let j=str.length-1;j>n;j--)
        {
            bag+=str[j]
        }
    }
    console.log(bag)
}
function runProgram(input)
{
    var str=input;
    check(str)
    // console.log(str)
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
```

```
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Majority element

Ended

Description

Given an array A having N non-negative integers. Find the element that occurs more than N/2 times. If no such element is there then print -1.

```
function check(N,arr){
    for(var i=0;N && i<=N/2;i++)
    {
        var count=1
        for( j=i+1;j<N;j++)
        {
            if(arr[j]==arr[i])
            {
                count++
            }
```

```
        }
        if(count>N/2)
        {
           return arr[i]
        }
     }
     return -1
}
function runProgram(input) {
   input=input.split("\n");
   var tc=+input[0];
   var line=1;

   for(var i=0;i<tc;i++){

       var N=+input[line]
       line++;
       var arr=input[line].split(" ").map(Number);
       line++;
    //   console.log(N,arr);
      console.log(check(N,arr))
   }
 }
 if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
```

```
 });
 process.stdin.on("end", function () {
  read = read.replace(/\n$/, "");
  read = read.replace(/\n$/, "");
  runProgram(read);
 });
 process.on("SIGINT", function () {
  read = read.replace(/\n$/, "");
  runProgram(read);
  process.exit(0);
 });
}
```

Day of the Week

Ended

Description

Given the current day, and a number N, find what day will it be after N days.

Note:Current day will be from the set ->{"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"}

```
function dayOfTheWeek(day, N) {
 var num=N%7
 var bag={
    Monday:1,
    Tuesday:2,
    Wednesday:3,
    Thursday:4,
    Friday:5,
    Saturday:6,
    Sunday:7
 };
```

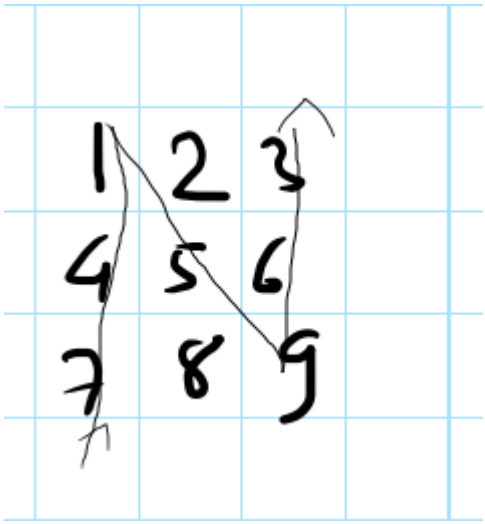
```
 for(key in bag)
{
   if(day==key){
      var value=bag[key];
      break;

   }
}
   var newValue=num+value;
   if(newValue>7){
      newValue=newValue%7;
   }
   for(key in bag){
      if(bag[key]==newValue){
         console.log(key)
         break;
      }
   }
}
```

N traversal

Ended

Description

You are given a matrix of size n x n. Find the**N**traversal of the matrix. Refer the follo wing figure for better understanding.




```
function nTraversal(matrix) {
  var bag="";
  for(var i=matrix.length-1;i>=0;i--){
     bag=bag+matrix[i][0]+" ";
  }
  for(var i=1;i<matrix.length;i++){
     bag=bag+matrix[i][i]+" ";
  }
  for(var i=matrix.length-2;i>=0;i--){
     bag=bag+matrix[i][matrix.length-1]+" ";
  }
  console.log(bag)
}
```

Matrix Traversal & Queries

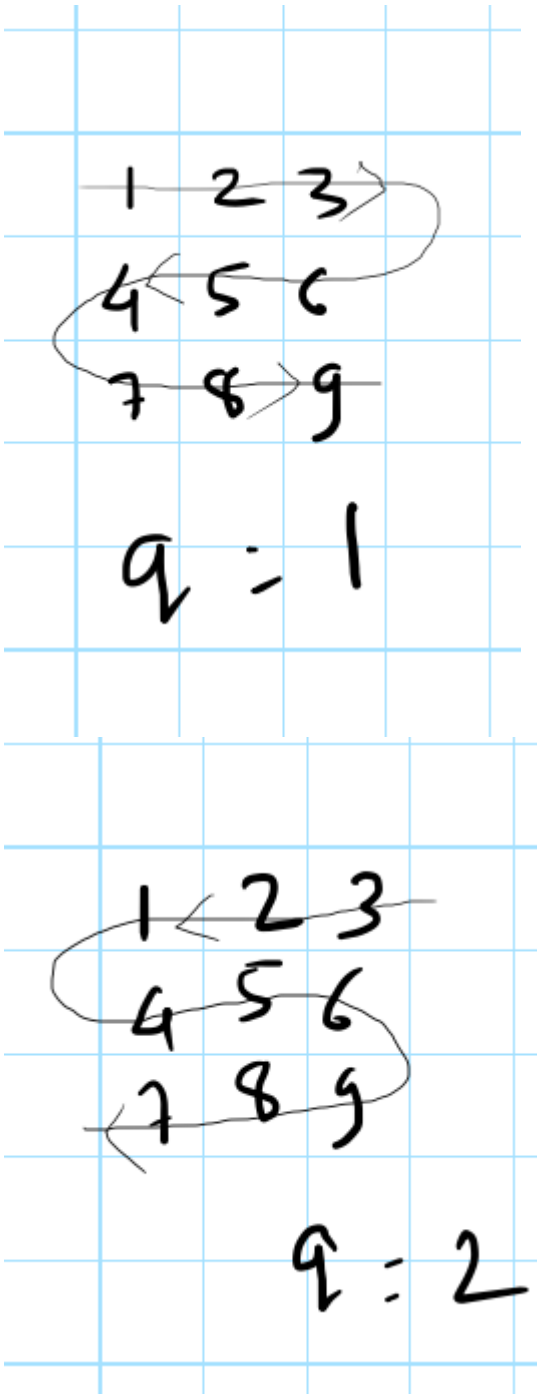
Ended

Description

You are given a matrix of size n x m , and two types of queries are to be performed o n this matrix. The two types of queries are

q = 1, for this type of query, the matrix is to be traversed, as shown in Fig. 1

q = 2, for this type of query, the matrix is to be traversed as shown in Fig. 2




```
function masaiTraversalAndQueries(N, M, q, arr){
   if(q==1){
      var bag="";
      for(var i=0;i<N;i++){
         if(i%2==0){
            for(var j=0;j<M;j++){
```

```
                bag=bag+arr[i][j]+" ";
            }
        }
        else{
            for(var j=M-1;j>=0;j--){
                bag=bag+arr[i][j]+" ";
            }
        }
    }
    console.log(bag);
  }
 else{
    var bag="";
    for(var i=0;i<N;i++){
        if(i%2==0){
            for(var j=M-1;j>=0;j--){
                bag=bag+arr[i][j]+" ";
            }
        }
        else{
            for(var j=0;j<M;j++){
                bag=bag+arr[i][j]+" ";
            }
        }
    }
    console.log(bag)
 }
}
```

https://oj.masaischool.com/contest/3533

Matrix Traversal & Queries

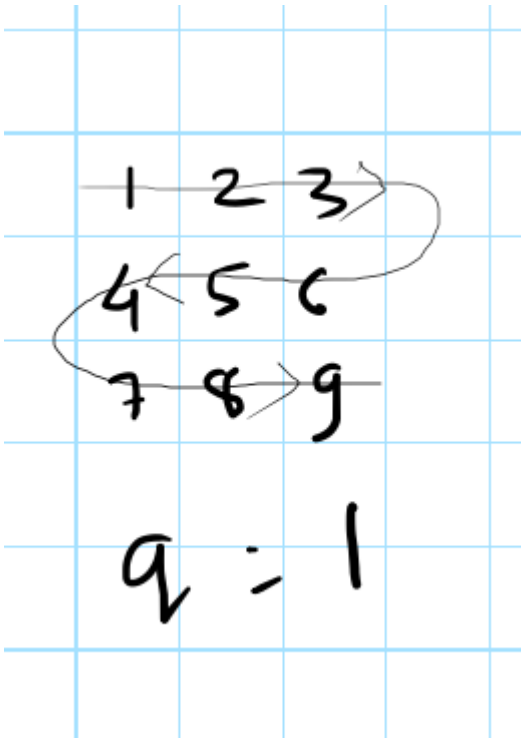
Ended

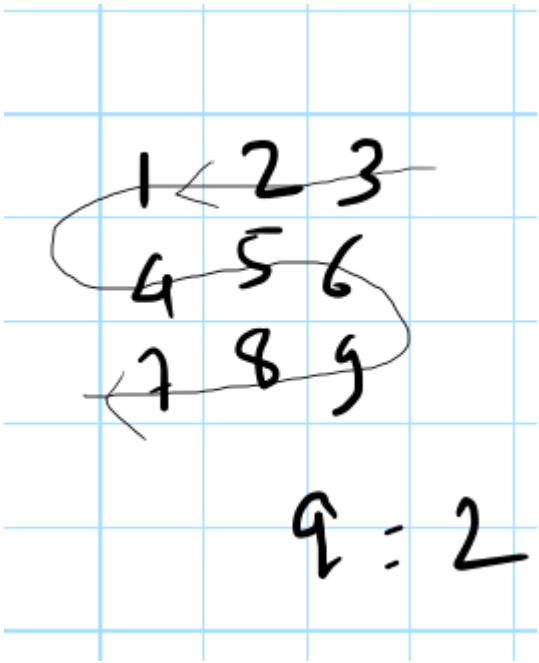
Description

You are given a matrix of size n x m , and two types of queries are to be performed o n this matrix. The two types of queries are

q = 1, for this type of query, the matrix is to be traversed, as shown in Fig. 1

q = 2, for this type of query, the matrix is to be traversed as shown in Fig. 2





Input

```
function masaiTraversalAndQueries(N, M, q, arr){
  //write code here
  if(q==1){
    var bag="";
    for(var i=0;i<arr.length;i++){
      if(i%2==0){
        for(var j=0;j<arr[i].length;j++){
          bag+=arr[i][j]+" ";
        }
      }
      else{
        for(var j=arr[i].length-1;j>=0;j--){
          bag+=arr[i][j]+" ";
        }
      }
    }
    console.log(bag);
  }
```

```
    else{
        var bag="";
        for(var i=0;i<arr.length;i++){
            if(i%2==0){
                for(var j=arr[i].length-1;j>=0;j--){
                    bag+=arr[i][j]+" ";
                }
            }
            else{
                for(var j=0;j<arr[i].length;j++){
                    bag+=arr[i][j]+" ";
                }
            }
        }
        console.log(bag);
    }



}
```

Exit the GridShow Editorial

Ended

Description

You are playing the famous Maze Runner Arcade Game. The game contains a maze which has values{'L', 'R', 'U', 'D'}, where L -> indicates that you move left, R -> indicates that you move right,

U -> indicates that you move up, while D -> indicates that you move down. The score is calculates as the number of moves in which you exit the grid. If you cannot exit the grid, that means your

scoreshould be returned as 0. Given a square matrix, denoting the maze, write a pro gram to calculate the score.

Note: You will always enter that the maze through the position (0,0).

```
function exitTheGrid(arr){
    let i =0;
    let j =0;
    var count=0;
    while(i<arr.length&&j<arr.length&&i>=0&&j>=0){
      if(arr[i][j]=="X"){


        return 0;
      }
      if(arr[i][j]=="R"){


        arr[i][j]="X";
        count++
        j++;
      }
     else if(arr[i][j]=="L"){
        count++
        arr[i][j]="X";
        j--;;
     }
     else if(arr[i][j]=="U"){
        count++
        arr[i][j]="X";
        i--;
     }
```

```
        else if(arr[i][j]=="D"){
            count++
            arr[i][j]="X";
            i++;
        }
    }
    return count;
 }


function runProgram(input){
  input=input.trim().split("\n");
    let test=+input[0];
    let line=1;
    for(let i=0;i<test;i++){
        let n=+input[line++];
        let arr=[];
        for(let j=0;j<n;j++){
            let a=input[line++].trim().split('');
            arr.push(a);
        }
        console.log(exitTheGrid(arr));

}

}




if (process.env.USER === "")
 {
```

```
runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");
```

```
runProgram(read);

process.exit(0);

});

}
```

Make Leaderboard

Ended

Description

You are given name and marks of N different students in a hackerrank contest. Your task is to write a program that makes leaderboard of the students under following co nditions:

- If two students get same marks they get same rank

- The student placed next to the same marks students will get the rank skipping the i ntermediate ranks.

Refer to the sample test case for better understanding

Note : You cannot use built-in sort function. Using that can lead to disqualification. W rite your own sorting algorithm

```
function board(name,marks){

  for(var i=0;i<name.length;i++)
  {
    for(var j=0;j<name.length-i-1;j++)
    {
      if(name[j]>name[j+1])
      {
        [name[j],name[j+1]]=[name[j+1],name[j]];
```

```
            [marks[j],marks[j+1]]=[marks[j+1],marks[j]];


        }


    }
}

for(var i=0;i<marks.length;i++)
{
    for(var j=0;j<marks.length;j++)
    {
        if(marks[j]<marks[j+1])
        {
            [marks[j],marks[j+1]]=[marks[j+1],marks[j]];
            [name[j],name[j+1]]=[name[j+1],name[j]];
        }
    }
}

 var arr=[1];
for(var i=1;i<marks.length;i++)
{
    if(marks[i]==marks[i-1])
    arr.push(arr[i-1]);
    else arr.push(i+1);
}

var bag="";
for(var i=0;i<marks.length;i++)
{
    bag=bag+arr[i]+" "+name[i]+"\n";
```

```
  }
  console.log(bag);
}
function runProgram(input)
{
 input=input.trim().split("\n");
 var tc=+input[0];
 var line=1,name=[],marks=[];
 for(var i=0;i<tc;i++)
 {
    var x=input[line].trim().split(" ");
    name.push(x[0]);
    marks.push(x[1]);
    line++;
 }
 marks=marks.map(Number);
//  console.log(name,marks);
 board(name,marks);
}
process.stdin.resume();
process.stdin.setEncoding("ascii");
let read="";
process.stdin.on("data",function(input)
{
  read+=input;
});
process.stdin.on("end",function(){
  read=read.replace(/\n$/,"");
  read=read.replace(/\n$/,"");
  runProgram(read)
});
```

Nick and Hacks

Ended

Description

Tom and Nick are good friends.Once Tom asked Nick exactly N rupees, but Nick has only 1 rupee in his bank account.

Nick wants to help his friend so he wrote two hacks First hack can multiply the amou nt of money he owns by 10,while the second can multiply it by 20. These hacks can be used any number of times.Can Nick help Tom with his hacks?

Input

```
function check(num,n){
   if(num==1){
      return true
   }
   if(n==num){
      return true
   }
   else if(n>num){
      return false;
   }
   return check(num,n*10)||check(num,n*20)
}



function runProgram(input){
 input=input.trim().split("\n")
 let n=+input[0]
 for(var i=1;i<=n;i++){
    var num=+input[i]
    var result=check(num,1)
```

```
    if(result==true){
        console.log("Yes")
    }else{
        console.log("No")
    }
 }

}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
    read += input;
```

```
});

process.stdin.on("end", function () {

 read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

 runProgram(read);
 });

 process.on("SIGINT", function () {

  read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Unique Subsets

Ended

Description

- You are given an array `A` of size `N`.
- Find out if there exists a subset of this array, such that the size of the subset is no lesser than K, and the number of distinct numbers in the subset is 1
- If such a subset exists, print `True`, else print `False`

NEED TO DO

https://oj.masaischool.com/contest/3542/problems

Spirals and DiagonalsShow Editorial

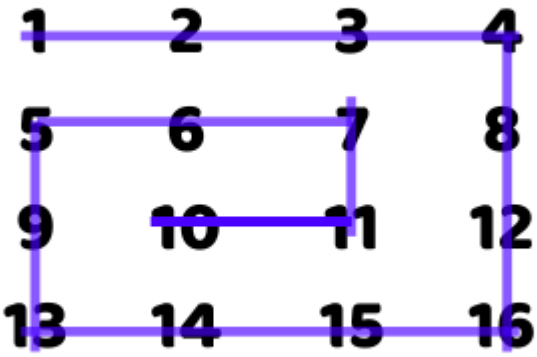
Ended

Description

You are given an integer n. The next line is an array A which contains n*n elements. The spiral traversal of a square matrix of dimension (n x n) results in the given array.

Calculate the sum of all elements which are on the diagonals of the square matrix.

The matrix traversal happens in the manner shown in the image below



NEED TO DO

Equilibrium Element

Ended

Description

Given an array A of N positive numbers. The task is to find the position where equilibrium first occurs in the array. Equilibrium position in an array is a position such that the sum of elements before it is equal to the sum of elements after it.

```
function equilibriumElement(N, arr){


   var res=Infinity;
   for(var i=0;i<N;i++){
      var left=0;
      for(var j=0;j<i;j++){
         left+=arr[j];
      }
      var right=0;
       for(let j=i + 1; j<N;j++){
          right+=arr[j]
       }
       if(left == right){
             res=i+1;
             break;
       }
   }
   if(res==Infinity){
      console.log("-1");
   }
   else{
      console.log(res);
   }
}
```

Product Of Array

Ended

Description

Given a array of N 32 bit integers. You need to find product of array for each i withou t considering ith element. where 0 <= i <= n-1 . See sample test case for better under standing.

You are not allowed to use division operator. It is given that product will fit in 32 bit in teger.

```
function  check(n,arr)
{
   let bag= ""
   var a=1
   for(var j=0;j<n;j++)
   {
      a*=arr[j]
   }
   let i=0;
   while(i<n)
   {
        bag+=(a/arr[i])+" "
        i++
   }
   console.log(bag)
}



function runProgram(input)
{
  var input = input.split("\n")
  var t = +input[0]
```

```
  var line = 1;
  for(var i = 0; i < t; i++)
  {
     var n = +input[line]
     line++
     var arr = input[line].split(" ").map(Number)
     line++
     check(n,arr)
  }
}

if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Next Greater Element

Ended

Description

Given an array of N elements, find the next greater element for each element in the a rray, print -1 if it does not exits. Refer to the sample I/O for better understanding

```
function check(n,arr){
   let ans=[];
   let stack=[];
   for(let i=n-1;i>=0;i--){
      while(stack.length!=0 && arr[i]>=stack[stack.length-1]){
         stack.pop();


      }
      if(stack.length==0){
         ans[i]=-1;
      }else{
         ans[i]=stack[stack.length-1]
      }
      stack.push(arr[i])
   }
   console.log(ans.join(' '))
}


function runProgram(input){
 input=input.trim().split("\n")
 var tc=+input[0]


 var line=1;
 for(var i=0;i<tc;i++){
```

```
    var n=+input[line];
    line++;
    var arr=input[line].trim().split(" ").map(Number)
   line++;
   check(n,arr)
 }

}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
  process.stdin.resume();

 process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
    read += input;
```

```
});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

https://oj.masaischool.com/contest/3551/problems

Factorial-Recursion

Ended

Description

The factorial of a positive integer N is the product of all positive integers less than or equal to n:

Given a number N your task is to write a program that calculates factorial of N

```
function runProgram(input){
    var N =+input[0]
    console.log(FactorialRecursion(N))
}
function FactorialRecursion(N){
    if(N==0){
        return 1
    }

        return N*FactorialRecursion(N-1)

}
if (process.env.USER === "") {
 runProgram(``);
} else {
 process.stdin.resume();
 process.stdin.setEncoding("ascii");
 let read = "";
 process.stdin.on("data", function (input) {
   read += input;
 });
 process.stdin.on("end", function () {
   read = read.replace(/\n$/, "");
   read = read.replace(/\n$/, "");
   runProgram(read);
 });
 process.on("SIGINT", function () {
   read = read.replace(/\n$/, "");
   runProgram(read);
   process.exit(0);
```

```
});
}
```

Fibonacci-Recursion

Ended

Description

In mathematics, the Fibonacci numbers, commonly denoted F(n) form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is

F (0) = 0 , F (1) = 1

and

F (n) = F (n − 1) + F (n − 2) ,

for n > 1.

Your task is to write a program that calculates n-th fibonacci numbers when you are provided with n

```
function runProgram(input){
   var N =+input
   var res = Fib(N)
   console.log(res)
}
function Fib(N){
   if(N==0 || N==1){
      return N
   }

      return Fib(N-1)+Fib(N-2)
```

```
}
if (process.env.USER === "") {
 runProgram(``);
} else {
 process.stdin.resume();
 process.stdin.setEncoding("ascii");
 let read = "";
 process.stdin.on("data", function (input) {
  read += input;
 });
 process.stdin.on("end", function () {
  read = read.replace(/\n$/, "");
  read = read.replace(/\n$/, "");
  runProgram(read);
 });
 process.on("SIGINT", function () {
  read = read.replace(/\n$/, "");
  runProgram(read);
  process.exit(0);
 });
```

Number of ways problems

Ended

Description

Sandhya is running up a staircase with N steps, and can hop(jump) either 1 step, 2 s teps or 3 steps at a time. You have to count, how many possible ways Sandhya can run up to the stairs.

```
function runProgram(input) {
```

```
    let N=Number(input)
    let ans=numberofways(N)
    console.log(ans)
}

function numberofways(N){
    if(N==0){
        return 1
    }
    if(N<0){
        return 0
    }
    else{
        return numberofways(N-1) +numberofways(N-2)+numberofways(N-3)
    }
}



if (process.env.USERNAME === "") {
    runProgram(``);
} else {
    process.stdin.resume();
    process.stdin.setEncoding("ascii");
    let read = "";
    process.stdin.on("data", function (input) {
        read += input;
    });
    process.stdin.on("end", function () {
        read = read.replace(/\n$/, "");
        read = read.replace(/\n$/, "");
        runProgram(read);
```

```
    });
    process.on("SIGINT", function () {
        read = read.replace(/\n$/, "");
        runProgram(read);
        process.exit(0);
    });
}
```

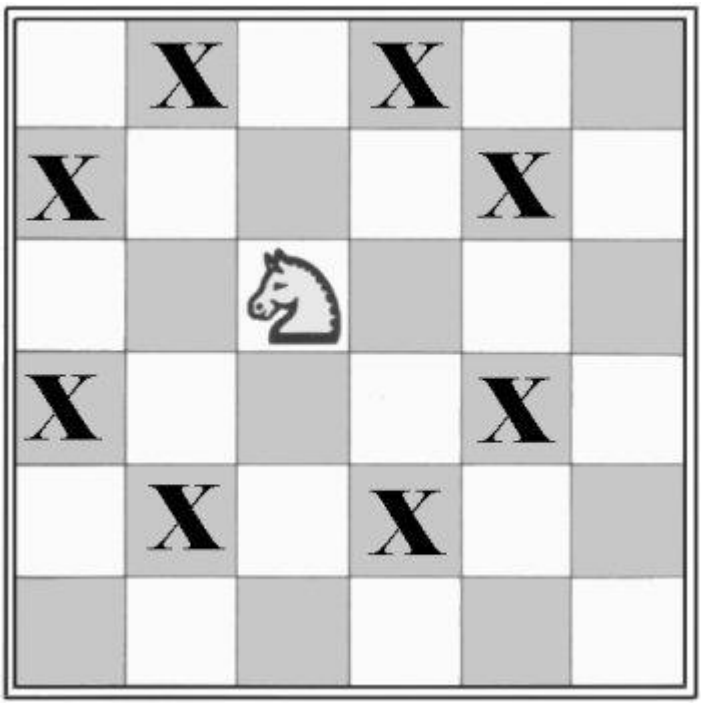
Follow The Knight

Ended

Description

You are very good at playing chess since childhood and you knew the rules very mu ch. You have a 10 X 10 chessboard and you want to explore all the possible square s on the board that the knight can be at in exactly N moves.It is placed at (i,j) coordin ate initially. For a 10X10 chessboard (1,1) will be the top left corner and (10,10) will b e the bottom right corner.

You can refer the following diagram, the knight can move to any of the squares mark ed as X in 1 move.



Input

Input Format

Input will consist of three space seperated integers i,j and N

Constraints

N < 10

Output

Print a single integer denoting the number of blocks on the chessboard that the knigh t can be at in exactly N moves.

```
function followknight(i,j,moves,obj)
{
  if(moves===0 && obj[`${i},${j}`]===undefined)
  {
    obj[`${i},${j}`]=1;
    return 1;
  }
  let count=0;
  if(moves>0)
  {
    let arr=[[1,2],[2,1],[1,-2],[2,-1],[-2,-1],[-1,-2],[-2,1],[-1,2]];
    for(let k=0;k<arr.length;k++)
    {
      let x=i+arr[k][0];
      let y=j+arr[k][1];
      if(x>=1 && x<=10 && y>=1 && y<=10)
      {
        count=count+followknight(x,y,moves-1,obj)
      }
    }
```

```
    }
    return count;
}
function runProgram(input)
{
    let[i,j,moves]=input.trim().split(" ").map(Number)
    let obj={}
    console.log(followknight(i,j,moves,obj));
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

https://oj.masaischool.com/contest/3580/problem/01

Number of Occurences in logn

Ended

Description

You are given n different numbers and an integer k. Write a program that finds numb er of times k is present in log(n) time complexity.

NOTE: YOU MUST NOT USE BRUTE FORCE SOLUTION

WE KNOW THAT YOU KNOW BRUTE FORCE SOLUTION AND WANT YOU TRY THE LOGN SOLUTION

USING BRUTE FORCE/LINEAR SEARCH IN THIS CASE WOULD LEAD TO DISQ UALIFICATION

Input

Input Format :

First line contains N and K

Second line contains N space separated sorted integers

```
function l(k,arr){
   let start=0;
   let end=arr.length-1;
   let res=-1


   while(start<=end){
      let mid=Math.floor(start+((end-start)/2));


      if(k==arr[mid]){
         res=mid;
         end=mid-1;
      }
      else if(arr[mid]>k){
         end=mid-1
```

```
        }
        else{
            start=mid+1;
        }
    }
    return res
}

function u(k,arr){
   let start=0;
    let end=arr.length-1;
    let res=-1

    while(start<=end){
        let mid=Math.floor(start+((end-start)/2));

        if(k==arr[mid]){
            res=mid;
            start=mid+1;
        }
        else if(arr[mid]>k){
            end=mid-1
        }
        else{
            start=mid+1;
        }
    }
    return res
}

function runProgram(input){
```

```
 input=input.trim().split("\n")
 let [n,k]=input[0].trim().split(" ").map(Number)
 let arr=input[1].trim().split(" ").map(Number)

 let lower=l(k,arr)
 let upper=u(k,arr)

 if(lower==-1){
    return 0
 }
 console.log((upper-lower)+1)
}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";
```

```
process.stdin.on("data", function (input)
{
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Lower bound in logn

Ended

Description

You are given n different numbers and an integer k. Write a program that finds lower bound of k in log(n) time complexity. Lower bound of a number k in a sorted list is the index of the first number which is same as k, in case the number is not present, print -1 (here the answer is given considering index to be starting from 0)

NOTE: YOU MUST NOT USE BRUTE FORCE SOLUTION.

WE KNOW THAT YOU KNOW BRUTE FORCE SOLUTION AND WANT YOU TO T RY THE LOGN SOLUTION.

USING BRUTE FORCE/LINEAR SEARCH, IN THIS CASE, WOULD LEAD TO DIS QUALIFICATION.

```
function check(arr,size,k){
    let start=0;
    let end=size-1;
   let res=-1;
    while(start<=end){

        let mid=Math.floor(start+((end-start)/2))
        if(k==arr[mid]){
            end=mid-1
            res=mid;
        }
        else if(k<arr[mid]){
            end=mid-1
        }

        else{
            start=mid+1
        }

    }
     return res
```

```
}
function runProgram(input){
 input=input.trim().split("\n")
 var [size,k]=input[0].split(" ").map(Number)
var arr=input[1].split(" ").map(Number)
console.log(check(arr,size,k))
}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });
```

```
process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Upper Bound in LogN

Ended

Description

You are given n different numbers and an integer k. Write a program that finds upper bound of k in log(n) time complexity. Upper bound of a number k in a sorted list is the index of the first number which is greater than k (here the answer is given considerin g index to be starting from 0)

-> Test cases are such that there is always one number greater than k

NOTE: YOU MUST NOT USE BRUTE FORCE SOLUTION

WE KNOW THAT YOU KNOW BRUTE FORCE SOLUTION AND WANT YOU TRY THE LOGN SOLUTION

USING BRUTE FORCE/LINEAR SEARCH, IN THIS CASE, WOULD LEAD TO DIS QUALIFICATION

```
function check(arr,size,k){
    let start=0;
    let end=size-1;
   let res=-1;
    while(start<=end){


        let mid=Math.floor(start+((end-start)/2))
        if(k>=arr[mid]){
            start=mid+1
            res=mid;
        }
        else{
            end=mid-1
        }


    }
     return res+1
}
function runProgram(input){
 input=input.trim().split("\n")
 var [size,k]=input[0].split(" ").map(Number)
var arr=input[1].split(" ").map(Number)
console.log(check(arr,size,k))
}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });
```

```
process.on("SIGINT", function () {

  read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Minimum in sorted & rotated array

Ended

Description

Given an array of length n, and it is sorted and rotated at some unknown point, find t he minimum element in it.

In a sorted & rotated array, we rotate an ascending order sorted array at some pivot unknown to you beforehand. So for instance, 1 2 3 4 5 might become 3 4 5 1 2.

Please note that the linear search approach with the time complexity of O(n) can eas ily be applied. But you are expected to solve it in O(log n) time complexity

```
function check(n,arr,low,high)
{

  while(low<high)
  {
    let mid = Math.floor(low+(high-low)/2);
    if (arr[mid]==arr[high])
      high--;
```

```
        else if(arr[mid]>arr[high])
            low=mid + 1;
        else
            high=mid;
    }
    console.log(arr[high]);
}

function runProgram(input){
 input=input.trim().split("\n")
 var n=+input[0]
 var arr=input[1].trim().split(" ").map(Number)
 var low=0;
 var high=n-1;
check(n,arr,low,high)
}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();
```

```
 process.stdin.setEncoding("ascii");

let read = "";

process.stdin.on("data", function (input)
{
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Search in sorted and rotated array

Ended

Description

Given a sorted(increasing order) array of length n and is rotated by some value befor ehand. Find the index of the target element K in the rotated array in O(logn) time. If not found print -1.

In a sorted & rotated array, we rotate an ascending order sorted array at some pivot unknown to you beforehand. So for instance, 1 2 3 4 5 might become 3 4 5 1 2.

Please note that the linear search approach with the time complexity of O(n) can eas ily be applied. But you are expected to solve it in O(log n) time complexity

Note: Array contains all distinct elements.

```
function check(n,p,arr,low,high){

  var mid = Math.floor((low + high) / 2);
  if(low>high){
    return -1;
  }

  if (arr[mid] == p){
    return mid;
  }
  if (arr[low] <= arr[mid]) {
    if (p >= arr[low] && p <= arr[mid]){
      return check(n,p,arr, low, mid - 1);

    }
```

```
        return check(n,p,arr, mid + 1, high);
    }
     if (p >= arr[mid] && p <= arr[high])
        return check(n,p,arr, mid + 1, high);

    return check(n,p,arr, low, mid - 1);

}



function runProgram(input){
 input=input.trim().split("\n")
 var [n,p]=input[0].trim().split(" ").map(Number)
 var arr=input[1].trim().split(" ").map(Number)
 var low=0;
var high=n-1;
 var res=check(n,p,arr,low,high)
if(res!=-1){
    console.log(res)
}else{
    console.log("-1")
}
}




if (process.env.USER === "")
 {
```

```
runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");
```

```
 runProgram(read);


 process.exit(0);


});


}
```

https://oj.masaischool.com/contest/3595/problems

Number of Occurences in logn

Ended

Description

You are given n different numbers and an integer k. Write a program that finds numb er of times k is present in log(n) time complexity.

NOTE: YOU MUST NOT USE BRUTE FORCE SOLUTION

WE KNOW THAT YOU KNOW BRUTE FORCE SOLUTION AND WANT YOU TRY THE LOGN SOLUTION

USING BRUTE FORCE/LINEAR SEARCH IN THIS CASE WOULD LEAD TO DISQ UALIFICATION

Input

```
function l(k,arr){
   let start=0;
   let end=arr.length-1;
   let res=-1


   while(start<=end){
      let mid=Math.floor(start+((end-start)/2));
```

```
        if(k==arr[mid]){
            res=mid;
            end=mid-1;
        }
        else if(arr[mid]>k){
            end=mid-1
        }
        else{
            start=mid+1;
        }
    }
    return res
}

function u(k,arr){
   let start=0;
    let end=arr.length-1;
    let res=-1

    while(start<=end){
        let mid=Math.floor(start+((end-start)/2));

        if(k==arr[mid]){
            res=mid;
            start=mid+1;
        }
        else if(arr[mid]>k){
            end=mid-1
        }
        else{
            start=mid+1;
```

```
        }
    }
    return res
}

function runProgram(input){
 input=input.trim().split("\n")
 let [n,k]=input[0].trim().split(" ").map(Number)
 let arr=input[1].trim().split(" ").map(Number)

 let lower=l(k,arr)
 let upper=u(k,arr)

 if(lower==-1){
     return 0
 }
 console.log((upper-lower)+1)
}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
```

```
  process.stdin.resume();

 process.stdin.setEncoding("ascii");

let read = "";

process.stdin.on("data", function (input)
{
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});
```

}

Kazama & Shinchan

Ended

Description

Shinchan & Kazama are classmates at Futaba Kindergarten. I- chan likes shinchan, more than Kazama. Kazama wants to prove to I-chan that he is smarter than Shinch an, so he tries to boast his knowledge about numbers in front of her. He takes an sor ted array of N integers, and asks I-chan for a number K. If the number exists in the ar ray, he will return the index where the number is present, else he returns the index w here it would be if it were inserted in order. This task seems to be very difficult for Ka zama, so he asks you for help. Help him, or else he will lose his friendship with I-cha n.

Input

The first line contains N, the size of the array. Next line contains N space separated i ntegers, indicating the values in the array.

Next line contains K, the value I-chan gave to search.

Constraints:

- `1 <= N <= 10^4`
- `-10^4<= array[i] <= 10^4`
- `nums`containsdistinctvalues sorted in**ascending**order.
- `-10^4<= K <= 10^4`

Output

Print a single integer denoting the position of K in the array, if present, else indicating the position where it would be if inserted in sorted order.

Sample Input 1

```
4

1 3 5 6
```

```
5
```

Sample Output 1

```
2
```

Sample Input 2

```
4

1 3 5 6

2
```

Sample Output 2

```
1
```

NEED TO NE DONE

https://oj.masaischool.com/contest/3604/problems

Insert at a specific position

Ended

Description

Given the pointer to the head node of a linked list and an integer to insert at a certain position, create a new node with the given integer as its data attribute, insert this nod e at the desired position and return the head node.

A position of 0 indicates head, a position of 1 indicates one node away from the head , and so on. The head pointer given may be null meaning that the initial list is empty.

Complete the functioninsertNodeAtPositionin the editor below. It must return a refere nce to the head node of your finished list.

insertNodeAtPosition has the following parameters:

- head: a SinglyLinkedListNode pointer to the head of the list
- data: an integer value to insert as data in your new node
- position: an integer position to insert the new node, zero-based indexing

Input

The first line contains an integer n, the number of elements in the linked list.

Each of the next nlines contains an integer SinglyLinkedListNode[i].data.

The next line contains an integer, the data of the node that is to be inserted.

The last line contains an integer position.

n <= 1000

list[i] <= 1000

Output

Print the updated Linked List

Sample Input 1

```
3
16
13
7
1
2
```

Sample Output 1

```
16 13 1 7
```

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};

// Complete the function below

function insertNodeAtPosition(head, data, position) {
    var node=new LinkedListNode(data)
    if(position==0){
        node.next=head;

        return node;
    }

  var node2=new LinkedListNode(data)
  node2.next=head;
  for(var i=0;i<position;i++)
 node2=node2.next
  node.next=node2.next
  node2.next=node

  return head;

}
```

Insert a node at the Head

Ended

Description

Given a pointer to the head of a linked list, insert a new node before the head. Retur n a reference to the new head of the list. The head pointer given may be null meanin g that the initial list is empty.

Complete the function insertNodeAtHead in the editor below.

insertNodeAtHeadhas the following parameter(s):

- LinkedListNode list: a reference to the head of a list
- data: the value to insert in the data field of the new node

Input

The first line contains an integer n, the number of elements to be inserted at the hea d of the list.

The next n lines contain an integer each, the elements to be inserted, one per functio n call.

1 <= n <= 1000

list[i] <= 1000

Output

For each case print the updated Linked List.

Sample Input 1

```
3
1
2
3
```

Sample Output 1

```
1
```

```
2 1

3 2 1
```

Insert a node at the Head

Ended

Description

Given a pointer to the head of a linked list, insert a new node before the head. Retur n a reference to the new head of the list. The head pointer given may be null meanin g that the initial list is empty.

Complete the function insertNodeAtHead in the editor below.

insertNodeAtHeadhas the following parameter(s):

- LinkedListNode list: a reference to the head of a list
- data: the value to insert in the data field of the new node

Input

The first line contains an integer n, the number of elements to be inserted at the hea d of the list.

The next n lines contain an integer each, the elements to be inserted, one per functio n call.

1 <= n <= 1000

list[i] <= 1000

Output

For each case print the updated Linked List.

Sample Input 1

```
3

1

2
```

```
3
```

Sample Output 1

```
1
2 1
3 2 1
```

Insert a node at the Tail

Ended

Description

You are given the pointer to the head node of a linked list and an integer to add to th e list. Create a new node with the given integer. Insert this node at the tail of the link ed list and return the head node of the linked list formed after inserting this new node . The given head pointer may be null, meaning that the initial list is empty.

You have to complete the`LinkedListNode insertAtTail(LinkedListNode head, int data)`method. It takes two arguments: the head of the linked list and the integer to in sert at the tail. You should not readany input from the stdin/console.

Input

The first line contains an integer n, denoting the elements of the linked list.

The next nlines contain an integer each, denoting the element that needs to be insert ed at the tail.

n <= 1000

list[i] <= 1000

Output

For each case print the updated Linked List

Sample Input 1

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};

// Complete the function below

function insertNodeAtTail(head, data) {
     var node=new LinkedListNode(data)
   var curr=head
    if(curr===null){
        return node;
    }
       while(curr.next!=null){
           curr=curr.next
       }
       curr.next=node;
       return head

}
```

Deleting a Node

Ended

Description

Delete the node at a given position in a linked list and return a reference to the head node. The head is at position 0. The list may be empty after you delete the node. In t hat case, return a null value.

Complete thedeleteNodefunction in the editor below.

deleteNodehas the following parameters:
-LinkedListNode pointer list:a reference to the head node in the list
-int position:the position of the node to remove

Input

The first line of input contains an integer n, the number of elements in the linked list.

Each of the next n linescontains an integer, the node data values in order.

The last line contains an integer,the position of the node to delete.

n <= 1000

list[i] <= 1000

Output

Print the updated Linked List

Sample Input 1

```
8
20
6
2
19
7
4
15
9
3
```

Sample Output 1

```
20 6 2 7 4 15 9
```

Language:
JavaScript

Reset to default code definition

Theme:
Terminal
Font Size: **20**

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};

// Complete the function below

function deleteNode(head, position) {

    if(position===0){
       head=head.next;
       return head;
   }
   let curr=head;
   for(let i=0;i<position-1;i++){
       curr=curr.next
   }
   curr.next=curr.next.next
   return head

}
```

Linked List Cycle

Ended

Description

Given`head`, the head of a linked list, determine if the linked list has a cycle in it.
There is a cycle in a linked list if there is some node in the list that can be reached ag ain by continuously following the`next`pointer. Internally,`pos`is used to denote the inde x of the node that the tail's`next`pointer is connected to.
Note-`pos`is not passed as a parameter.
Complete the function below. No need to take any input, just return if the given linked list has a cycle in it or not. Return`true`if there is a cycle in the linked list. Oth erwise, return`false`.
Note- All the nodes are different

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};
// Complete the function below
var hasCycle = function(head) {
    var slow=head;
var fast=head;
while(fast!==null && fast.next!==null){
    slow=slow.next
    fast=fast.next.next;
    if(slow==fast){
        return true
    }

}
return false
};
```

Middle Node

Ended

Description

Given a non-empty, singly linked list with the head node`head`, return a middle node of the linked list.
If there are two middle nodes, return the second middle node.

Complete the functionbelow, no need to take any input.

Input

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};

var middleNode = function(head) {
    if(head==null){
        return null;
    }
    let slow=head;
    let fast=head;
    while(fast!=null && fast.next!=null){
        slow=slow.next;
        fast=fast.next.next;
    }
    return slow.data

};
```

Nth node from the end

Ended

Description

Given a linked list consisting of nodesand given a number N. The task is to find the n th nodefrom the end of the linked list.

No need to take any input, just complete the function below and return the nth nod e.

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};

function nth_node(head,n){
     let len=0;
   let curr=head;
   while(curr!=null){
       curr=curr.next;
       len++;}
   if(len<n){
       return;}
   curr=head;
   for(var i=0;i<len-n;i++){
       curr=curr.next
   }
   return curr.data

}
```

Reverse the Linked List

Ended

Description

Given the pointer to the head node of a linked list, change thepointers of the nodes s o that their order is reversed. The head pointer given may be null meaning that the in itial list is empty.

Complete thereversefunction in the editor below.

reversehas the following parameter:

- LinkedListNode pointer head:a reference to the head of a list

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
}

// Complete the function below

function reverse(head) {

   var next=null;
   var curr=head;
   var prev=null;
   while(curr!==null){
       next=curr.next;
       curr.next=prev;
       prev=curr;
       curr=next
```

```
  }
  var node=prev;
  return node;

}
```

Different Ways To Buy CandiesShow Editorial

Ended

Description

Betty loves candies, so she went to the store to buy some. She has some coins and she wants to spend all of them. The store has N distinct candies each having some c ost and there are infinite amount of each candy in the store. Betty can choose the sa me candy any number of times as long as she have coins greater than or equal to th e cost of that candy. Find and display the unique combinations in which she can buy candies such that the sum of cost of all the candies is equal to number of coins she h as.

Two combinations are unique if the frequency of at least one of the chosen candies i s different.

The input array is sorted in ascending order.

Input

NEED TO BE DONE

Betty Buys A PresentShow Editorial

Ended

Description

Betty and Archie are best friends and since Archie's birthday is in one week, Betty w ants to buy a present for him. She goes to the gift shop and selects a gift of price P. Betty has 9 coins each of value from 1 to 9 respectively. Find out the different in whic h Betty can pay for the gift such that she can only use K coins.

Input

```
let new_s=[];
let out=[];
function check(n,s,p,index){
    if(n==0){
        new_s.push(out.join(" "))
        return;
    }
    for(let i=index;i<=s.length;i++){
        out.push(i)
        check(n-1,s,p,i+1)
        out.pop()
    }
}

function runProgram(input){
 input=input.trim().split(" ")
 let p=+input[0]
 let n=+input[1]
 let s=[];
 for(let i=1;i<=9;i++){
     s.push(i)
 }
 check(n,s,p,1)
 let flag=false;
 for(var i=0;i<new_s.length;i++){
     let s1=new_s[i].trim().split(" ").map(Number);
     let sum=0;
```

```
    for(key in s1){
        sum+=+s1[key]
    }
    if(sum==p){
        flag=true;
        console.log(s1.join(" "))
    }
 }
 if(flag==false){
    console.log('-1')
 }

}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
    read += input;
```

```
});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

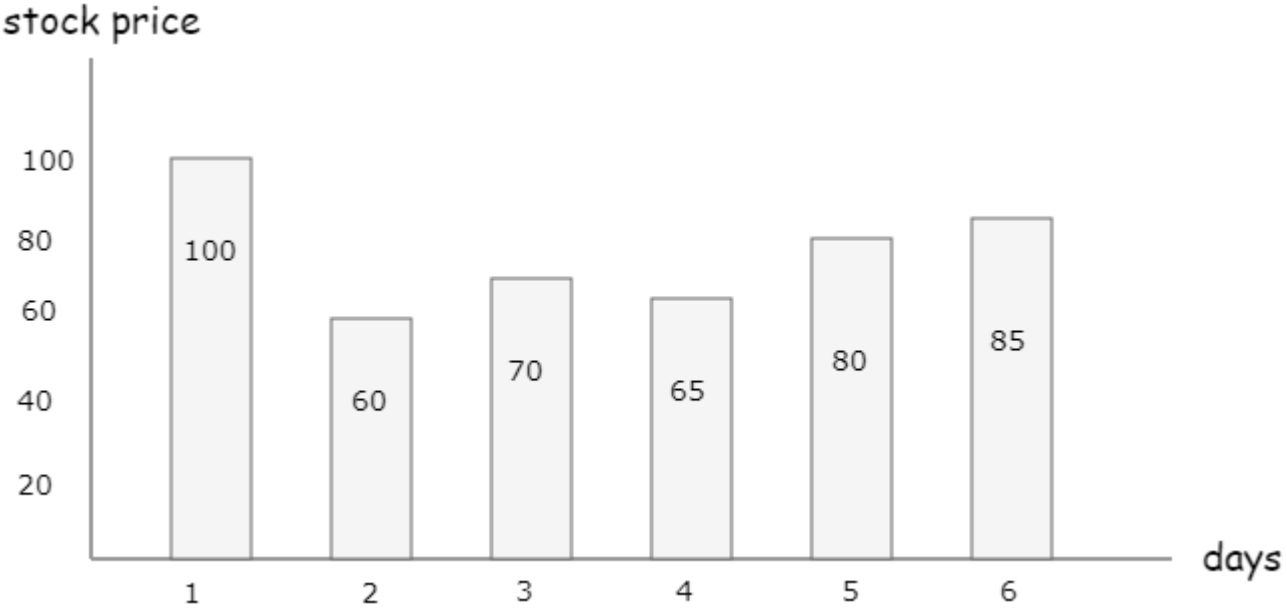
Solve a financial problem

Ended

Description

Given a list of prices of a single stock for N number of days, find stock span for each day. Stock span is defined as a number of consecutive days prior to the current day

when the price of a stock was less than or equal to the price at current day.



For the first day, span is always 1. In the example we can see that for day 2 at 60, th ere is no day before it where the price was less than 60. Hence span is 1 again. For day 3, the price at day 2 (60) is less than 70, hence span is 2. Similarly, for day 4 an d day 5. Remember days should be consecutive, that's why the span for day 4 is 1 – even though there was a day 2 where the price was less than 65.

Input

Input Format

First line contains an integer T specifying the number of test cases.

First line of each test case contains the value of N

Second line of each test case contains N space separated integers which are the sto ck price for N days

Constraints

```
function check(n,arr,s){
    var res="";
    s[0]=1;
     for (let i = 1; i < n; i++) {
            let count = 1;
            while ((i - count) >= 0 && arr[i] >= arr[i - count]) {
                count += s[i - count];
```

```
            }
            s[i] = count;
          }

      for(var i=0;i<n;i++){
        res=res+s[i]+" ";
      }
 console.log(res)
}


function runProgram(input){
 input=input.trim().split("\n")
 var tc=+input[0]
 var line=1;

 for(var i=0;i<tc;i++){
    var num=+input[line]
    line++;
    var arr=input[line].trim().split(" ").map(Number)
    line++;
    check(num,arr,s=[])
 }

}




if (process.env.USER === "")
 {

runProgram(``);
```

```
 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

 runProgram(read);

 process.exit(0);

});

}
```

Masai Coding Competition

Ended

Description

There is a coding Tournament where 4 clubs are going to compete. Since the team s election is very critical in this competition,Rohit asked Harshit to help him in the team selection process.

There is a long queue of students from four clubs. Each student of a club have a diff erent roll number. Whenever a new student will come, he will search for his clubmate from the end of the queue. As soon as he will find any of the club-mate in the queue, he will stand behind him, otherwise he will stand at the end of the queue. At any mo ment Harshit will ask the student, who is standing in front of the queue, to come and give his name and Harshit will remove him from the queue. There are Q operations o f one of the following types:

E a b: A new student of club a whose roll number is b will stand in queue according t o the method mentioned above.

D: Harshit will ask the student, who is standing in front of the queue, to come and put his name and Harshit will remove him from the queue

Input

```
function runProgram(input)
{
  input = input.trim().split(/[\r\n]+/);
  let size = +input[0].trim();
  let c1 = [];
  let c2 = [];
  let c3 = [];
  let c4 = [];

  let q= [];
```

```
for (let i = 1; i < input.length; i++)
{
  let current = input[i].trim().split(" ");
  if (current[0] === "E")
  {
    if (+current[1] === 1)
    {
      if (q.includes(1))
      {
        c1.push([current[1], current[2]]);
      }
      else
      {
        c1.push([current[1], current[2]]);
        q.push(1);
      }
    }
    else if (+current[1] === 2)
    {
      if (q.includes(2))
      {
        c2.push([current[1], current[2]]);
      }
      else
      {
        c2.push([current[1], current[2]]);
        q.push(2);
      }
    }
    else if (+current[1] === 3)
    {
      if (q.includes(3))
      {
        c3.push([current[1], current[2]]);
```

```
    }
    else
    {
      c3.push([current[1], current[2]]);
      q.push(3);
    }
  }
  else
  {
    if (q.includes(4))
    {
      c4.push([current[1], current[2]]);
    }
    else
    {
      c4.push([current[1], current[2]]);
      q.push(4);
    }
  }
}
else
{
  if (q[0] === 1)
  {
    let temp = c1.shift();
    console.log(temp.join(" "));
    if (c1.length === 0)
    {
      q.splice(0, 1);
    }
  }
  else if (q[0] === 2)
  {
    let temp = c2.shift();
    console.log(temp.join(" "));
```

```
        if (c2.length === 0)
        {
          q.splice(0, 1);
        }
      }
      else if (q[0] === 3)
      {
        let temp = c3.shift();
        console.log(temp.join(" "));
        if (c3.length === 0)
        {
          q.splice(0, 1);
        }
      }
      else
      {
        let temp = c4.shift();
        console.log(temp.join(" "));
        if (c4.length === 0)
        {
          q.splice(0, 1);
        }
      }
    }
  }
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
```

```
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Super Digit

Ended

Description

We define super digit of an integer N using the following rules:

- If N has only 1 digit, then its super digit is N.
- Otherwise, the super digit of N is equal to the super digit of the sum of the digits of N.

Given an integer, find the super digit of the integer.

Input

```
function Super(num)
{
    if(num.length==1)
    {
        console.log(num)
        return
    }
    let sum=0;
```

```
    for(let i=0;i<num.length;i++)
    {
        sum+=(Number)(num[i])
    }
    Super((String)(sum))
}
function runProgram(input)
{
   input = input.trim().split('\n')
   let n=+input[0]
   for (let i=1;i<=n;i++)
   {
       let num = input[i];
       Super(num)
   }
}
if (process.env.USER === "") {
   runProgram(``);
 } else {
   process.stdin.resume();
   process.stdin.setEncoding("ascii");
   let read = "";
   process.stdin.on("data", function (input) {
     read += input;
   });
   process.stdin.on("end", function () {
     read = read.replace(/\n$/, "");
     read = read.replace(/\n$/, "");
     runProgram(read);
   });
   process.on("SIGINT", function () {
     read = read.replace(/\n$/, "");
     runProgram(read);
     process.exit(0);
  });
```

}

https://oj.masaischool.com/contest/3639/problems

Push, Pop and Top

Ended

Description

There is a stack of integers which is currently empty. You are given an integer n and there are n operations that you need to perform on the stack.

The next n line contains one of the following 3 operations:

1 x : Push x to the top of the stack.

2 : Pop an element from the top of the stack. If the stack is empty, do nothing.

3 : Print the top element of the stack (if stack is empty, print "Empty!" (without quotes)).

For better understanding, read sample test case explanation

```
function check(arr,newarr){
    newarr.push(arr[1])
    return (newarr)
}
function check2(num,newarr){
    if(num!=3){
        newarr.pop(newarr[1])
    }
    else if(num==3){
        if(newarr.length===0){
            console.log("Empty!");
        }
```

```
        else{
            console.log(newarr[newarr.length-1]);
        }
    }
}
```

```
function runProgram(input){
  input=input.trim().split("\n");
  //console.log(input);
  var tc=+input[0];
var newarr=[];
for(var i=1;i<=tc;i++){
    if(input[i].length>1){
        var arr=input[i].trim().split(" ").map(Number)
        var s=check(arr,newarr)
    }
    else if(input[i].length==1){
        var num=+input[i]
        check2(num,newarr)
    }
}

}
```

```
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
```

```
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Smaller Neighbour Element

Ended

Description

Given an array, find the nearest smaller element G[i] for every element A[i] in the arr ay such that the element has an index smaller than i.

Mathematically,

G[i] for an element A[i] is an element A[j] such that

j is maximum possible AND

j < i AND

A[j] < A[i]

Note: Elements for which no smaller element exist, consider next smaller element as -1.

Input

```
function next(n,arr){
    let res=[];
    let stack=[];
    for(let i=0;i<n;i++){
        while(stack.length!==0 && stack[stack.length-1]>=arr[i]){
            stack.pop();
        }
        if(stack.length==0){
            res[i]=-1;
        }else{
            res[i]=stack[stack.length-1]
        }
        stack.push(arr[i])
    }
    console.log(res.join(" "))
}



function runProgram(input){
 input=input.trim().split("\n")
 var n=+input[0]
 var arr=input[1].trim().split(" ").map(Number)
 next(n,arr)
 //console.log(arr)

}




if (process.env.USER === "")
 {
```

```
runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

 runProgram(read);

 process.exit(0);

});
```

}

Next Greater Element

Ended

Description

Given an array of N elements, find the next greater element for each element in the a rray, print -1 if it does not exits. Refer to the sample I/O for better understanding

Input

The first line contains T, the number of test cases.

The first line of each test case contains N, the number of elements in the array. The next line contains N space separated integers denoting the elements of the array

Constraints

1 <= T <= 10

1 <= N <= 10^5

0 <= A[i] <= 10000

Output

For each test case, print N space separated integers, denoting the next greater elem ent for each element, on a new line.

```
function check(size,arr){
    var next=[];
    var stack=[];
    for(var i=size-1;i>=0;i--){
        while(stack.length!==0 && arr[i]>=stack[stack.length-1]){
            stack.pop();
        }
        if(stack.length==0){
```

```
            next[i]=-1;
        }
        else{
            next[i]=stack[stack.length-1];
        }
        stack.push(arr[i]);
    }
    console.log(next.join(" "));
}

function runProgram(input){
    input=input.trim().split("\n");
    var tc=+input[0];
    var line=1;
    for(var i=0;i<tc;i++){
        var size=+(input[line]);
        line++;
        var arr=input[line].trim().split(" ").map(Number);
        line++;
        check(size,arr);
    }


}




if (process.env.USER === "")
 {

runProgram(``);

 }
```

```
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

 runProgram(read);

 process.exit(0);

});

}
```

Reduce String

Ended

Description

Given a string of lowercase characters in range ascii[‘a’..’z’].

You can perform one operation on this string in which you can selects a pair of adjac ent lowercase letters that match, and delete them.

For instance, the string aab could be shortened to b in one operation.

Your task is to delete as many characters as possible using this method and print the resulting string. If the final string is empty, print "Empty String" (without quotes).

Please note that characters can be deleted only if they form a pair and are same(i.e f rom aaa we can only delete 2 a's and will be left with a single a).

I know there exists a simple implemented Stringentation based solution of this questi on, but please try to come up with an approach that uses stack data structure to solv e the purpose

Input

Input Format

First and the only line contains string

Constraints

Length of string < 1000

Output

Output Format

If the final string is empty, print Empty String; otherwise, print the final non-reducible string.

Sample Input 1

```
aaabccddd
```

Sample Output 1

```
abd
```

Hint

```
function check(arr){
    var stack=[];
   for(var i=0;i<arr.length;i++){
       if(arr[i]===stack[stack.length-1]){
           stack.pop()
       }
       else{
           stack.push(arr[i])
       }
   }
   if(stack.length!==0){
       console.log(stack.join(""))
   }
   else{
       console.log("Empty String")
   }




}

function runProgram(input){
    var arr=input.trim();
    check(arr);
    //console.log(arr);

}
```

```
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Selection Sort Problem

Ended

Description

You are given an array of N unsorted numbers. Your task is to write SELECTION SORT such that numbers present in the array gets sorted.

USING ANY OTHER ALGORITHM OR USING IN BUILT SORT FUNCTION WOULD LEAD TO YOUR DISQUALIFICATION

Input

Input Format:

First line of input contains N

Second line of input contains N numbers

Constraints:

N < 500

Output

Output the numbers given after sorting it in increasing order

Sample Input 1

```
5

3 5 0 9 8
```

Sample Output 1

```
0 3 5 8 9
```

```
function solve(N,arr){
    //write code here
    for(var i=0;i<N;i++){
        var min=i;
        for(var j=i;j<N;j++){
            if(arr[j]<arr[min]){
            min=j;
        }
    }
    [arr[i],arr[min]]=[arr[min],arr[i]]



}
```

```
console.log(arr.join(" "));
}
```

Bubble Sort Problem

Ended

Description

You are given an array of N unsorted numbers. Your task is to write BUBBLE SORT such that numbers present in the array gets sorted.

USING ANY OTHER ALGORITHM OR USING IN BUILT SORT FUNCTION WOUL D LEAD TO YOUR DISQUALIFICATION

Input

Input Format:

First line of input contains N

Second line of input contains N numbers

Constraints:

N < 500

Output

Output the numbers given after sorting it in increasnig order

```
function solve(N,arr){
    //write code here
    for(var i=0;i<arr.length-1;i++){
        for(var j=0;j<arr.length-i-1;j++){
            if(arr[j]>arr[j+1]){
                [arr[j],arr[j+1]]=[arr[j+1],arr[j]]
            }
        }
    }
```

```
    console.log(arr.join(" "));
}
```

https://oj.masaischool.com/contest/3662/problems

Fizzz buzzz

Ended

Description

- You are given a number stored in a variable with the name `num`
- For all numbers in the range `[1,num], including num`

```
1. If the number is divisible by 3 and 5 both, print FizzBuzz

2. Else If the number is divisible by 3, print "Fizz", without quotes

3. Else If the number is divisible by 5, print "Buzz", without quotes

4. Else, print the number as it is
```

```
function fizzBuzz(num) {
    // Write code here
    for(var i=1;i<=num;i++){
        if((i%3===0) && (i%5===0)){
            console.log("FizzBuzz");
        }
        else if(i%3===0){
            console.log("Fizz");

        }
        else if(i%5===0){
            console.log("Buzz");
        }
        else{
            console.log(i);
```

```
        }
    }

}
function runProgram(input){
input=input.trim().split("\n")
var tc=+input[0]
var line=1;

for(var i=0;i<tc;i++){
    var num=+input[line].trim()
    line++;
    fizzBuzz(num)
}



}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");
```

```
let read = "";

process.stdin.on("data", function (input)
{
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Identify Prime

Ended

Description

- You are given a number stored in a variable with the name `num`

- Check if the number is a prime number or not
- If the value stored in `num`, is a prime number print `Yes`, else print `No`

```
Note : A prime number is a number, that is divisible by only 1 and the
number itself
```

Input

The first and the only line of the input contains the value stored in `num`

Output

- If the value stored in `num`, is a prime number print `Yes`, else print `No`

```
function identifyPrime(num) {
    for(var i=2;i<=num;i++){
        if(num%i===0)
            break;
    }
        if(num==i){
            console.log("Yes");
        }
        else{
            console.log("No");
        }

}
```

Binary Equivalent - Recursive

Ended

Description

Given an integer n, you need to find out its binary representation using recursion.

Here multiple test cases exist and the expected time complexity is - O(t*logn) where i s t is the number of test cases.

Input

The first line of the input contains one integer t (1 ≤ t ≤ 10) — the number of test cas es. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100000) — the nu mber.

Output

For each test case, print the answer: The binary representation of the integer.

Sample Input 1

```
2
15
128
```

Sample Output 1

```
1111
10000000
```

```
function check(n,bag){
    if(n==0){
        let arr=bag.split("")
        arr.reverse()
        console.log(arr.join(""))
        return

    }
    bag=bag+(n%2)
    return check(parseInt(n/2),bag)
}

function runProgram(input){
 input=input.trim().split("\n")
 let tc=+input[0]
```

```
 let line=1;
 let bag="";
 for(var i=0;i<tc;i++){
     let n=+input[line]
     line++;
     check(n,bag)
 }

}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {
```

```
  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Anagram Detector!

Ended

Description

An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once. For example, the word anagram can be rearranged into nag a ram.

Given 2 phrases, write a program that detects if both are anagrams of each other.

If both are anagrams, print "True"

Else print "False"

Input

Input Format :

First line of input contains first phrase

Second line of input contains second phrase

```
function runProgram(input){
input=input.split("\n")
let s1=input[0].split(" ").join("")
 let s2=input[1].split(" ").join("")



 console.log(check(s1,s2))

}
function check(s1,s2){
    let n1=s1.split("").sort();
    let n2=s2.split("").sort();


    for(var i=0;i<n1.length;i++){
        if(n1[i]!==n2[i]){
            return "False"
        }

    }

    return "True"

}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

 runProgram(read);
```

```
 process.exit(0);

});

}
```

Power of 2

Ended

Description

Given an integer N, check if the number is power of 2 or not.

Note : You may not use any inbuilt function.

Input

The first line contains T, the number of test cases.

The first and the only line of each test case contains N, the number to be checked.

Constraints :

- `-2^31<= n <= 2^31- 1`

Output

Print "True", if the number is a power of 2, else print "False".

```
function runProgram(input){
    input = input.trim().split("\n")
   var tc =+input[0]
   var line = 1
   for(var i=0;i<tc;i++){
       var num =+input[line]
       line++
     console.log(Power(num))
   }
}
function Power(num){
```

```
    if(num === 1){
      return "True";
    }
    if(num % 2 !== 0){
      return "False";
    }
    return Power(num / 2);
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

is it power of 3

Ended

Description

Given a number x, find whether it is a power of 3 or not.

Input

1<=T<=10

1<=N<=1e18

Output

output YES or NO based on the question for each test case

Sample Input 1

```
3
2
3
4
```

Sample Output 1

```
NO
YES
NO
```

```
function runProgram(input){
  input = input.trim().split("\n")
    var tc =+input[0]
    var line = 1
    for(var i=0;i<tc;i++){
        var num =+input[line]
        line++
      console.log(Power(num))
    }
}
function Power(num){
    if(num === 1){
      return "YES"
   }
   if(num % 3 !== 0){
```

```
      return "NO"
    }
    return Power(num/3);
}
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Remove duplicates

Ended

Description

Given an array of n integers, the array is sorted. You have to remove the duplicates, print only unique elements, do it in place. i.e O(1) space

Input

1<=T<=10

1<=N<=100000

1<=Ai<=100000

Output

output a single integer x, i.e the number of unique elements in the array and in the n ext line print the x unique elements.

NOTE: Do it inplace

Sample Input 1

```
2
3
1 1 2
4
1 1 3 3
```

Sample Output 1

```
2
1 2
2
1 3
```

```
function check(n,arr){
  var bag={};
  for(var i=0;i<n;i++){
     if(bag[arr[i]]==undefined){
        bag[arr[i]]=1
     }
     else{
        bag[arr[i]]=bag[arr[i]]+1
     }
```

```
   }
   return bag
}
function check2(bag){
     var res="";
     var count=0;
    for(key in bag){
      res=res+key+" ";
    count++;
   }
   console.log(count)
    console.log(res)
}
function runProgram(input){
 input=input.trim().split("\n")
 var tc=+input[0]
 var line=1;
 for(var i=0;i<tc;i++){
     var n=+input[line];
     line++;
     var arr=input[line].trim().split(" ").map(Number)
     line++;
    var bag= check(n,arr)
     check2(bag)


 }

}




if (process.env.USER === "")
```

```
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

 runProgram(read);

 process.exit(0);
```

});

}

https://oj.masaischool.com/contest/3686/problems

The Peak Point

Ended

Description

Given an array of size n, which is strictly increasing and then strictly decreasing in order. Find out the index (0-based) which is the peak of the array.

Note that the extremities are not considered as peak and there exists a peak which is not at index 0 or at n-1.

Input

The first line of the input contains one integer t (1 ≤ t ≤ 10) — the number of test cases. Then t test cases follow.

The first line of each test case contains a single integer n (3 ≤ n ≤ 100000).

The second line of the test case contains n integers (1 ≤ Ai ≤ 100000) and they are all distinct in nature.

Output

For each test case, print the answer: The peak index.

Sample Input 1

```
2

3

10 20 11

5

1 3 6 5 4
```

Sample Output 1

```
1

2
```

```
function check(n,arr){
    if(n==0){
        return 0
    }
    if (arr[0] >= arr[1]) {
        return 0;
    }
        if (arr[n - 1] >= arr[n - 2]) {
            return n - 1;
        }
        for(var i=0;i<n;i++){
           if (arr[i] >= arr[i - 1] && arr[i] >= arr[i + 1]) {
               return i
           }
        }

}

function runProgram(input){
 input=input.trim().split("\n")
 var tc=+input[0]
 var line=1;
 for(var i=0;i<tc;i++){
     var n=+input[line];
     line++;
     var arr=input[line].trim().split(" ").map(Number)
     line++;
     //console.log(n,arr)
     console.log(check(n,arr))
```

```
 }

}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
```

```
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Rotate Elements

Ended

Description

Given a n by n matrix. You have to rotate the elements of each ring of the matrix in t he clockwise direction one place.

Input

Input Format

First line will contain a single number n

Next n lines will contain the matrix

Constraints

n<=1000

Elements of the matrix <=10000

Output

You have to display the rotated matrix

Sample Input 1

```
4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
```

Sample Output 1

```
1 1 2 3
1 2 2 4
1 3 3 4
2 3 4 4
```

```
function runProgram(input){
    input = input.trim().split('\n');
    var x=input[0].trim().split(' ').map(Number);
    var n=+(x[0]);
    let mat=[];
    var line=1;
    for(var z=0;z<n;z++){
        mat.push(input[line++].trim().split(' ').map(Number));
    }
    rotatematrix(n, mat);
}
function rotatematrix(n, mat)
{
    let m=n;
    let row = 0, col = 0;
    let line, bag;
    while (row < m && col < n)
```

```
{
    if (row + 1 == m || col + 1 == n)
    {
       break;
    }
    line = mat[row + 1][col];
    for(let i = col; i < n; i++)
    {
        bag = mat[row][i];
        mat[row][i] = line;
        line = bag;
    }
    row++;
    for(let i = row; i < m; i++)
    {
        bag = mat[i][n - 1];
        mat[i][n - 1] = line;
        line = bag;
    }
    n--;
    if (row < m)
    {
        for(let i = n - 1; i >= col; i--)
        {
            bag = mat[m - 1][i];
            mat[m - 1][i] = line;
            line = bag;
        }
    }
    m--;
    if (col < n)
    {
        for(let i = m - 1; i >= row; i--)
        {
            bag = mat[i][col];
```

```
                mat[i][col] = line;
                line = bag;
            }
        }
        col++;
    }

    for(let i = 0; i < mat.length; i++)
    { let result="";
        for(let j = 0; j <mat.length; j++){
            result=result+mat[i][j]+" ";
        }
        console.log(result);
    }
}

if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
```

}

Shop Soap

Ended

Description

Piyush goes to buy soap from a shop.The shop contains N soaps. The prices of soa p are given in the form of an array A. The price of ith soap is A[i]. Now Piyush has q queries, in each query he wants to know the number of soaps that have price less th an the given amount M.

Input

Input Format :

First line contains integer N total number of soaps available in the shop.

Second line contains N space separated integers.

Third line contains Q number of queries.

Each of the next Q lines contain integer M.

Constraints :

1 ≤ N ≤ 100000

1 ≤ A[i] ≤ 1000000000

1 ≤ Q ≤ 100000

1 ≤ M ≤ 100000

Output

For each query output number of soaps having price less than M for that query.

Sample Input 1

```
5

1 4 10 5 6
```

```
4
2
3
5
11
```

Sample Output 1

```
1
1
2
5
```

```
function shop(N,arr,k){

    let lowerbound =0;
    var i =0;
    var j =N-1;
    while(i<=j){
var mid = i+(Math.floor((j-i)/2));
if(arr[mid]<k){
i=mid+1;
}else{
lowerbound=mid;
j=mid-1;
}
    }
    if(arr[N-1]<k){
        console.log(N)
    }else{
      console.log(lowerbound)
    }
```

```
  }


function runProgram(input){
    input=input.trim().split("\n");
    let N =+input[0];
    let arr =input[1].trim().split(" ").map(Number).sort((a,b)=>a-b);
    let tc =+input[2];
    let line =3;

    for(let i =0;i<tc; i++){
        var k =+input[line++]
        shop(N,arr,k);
    }
}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
```

```
{
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Masai Uniqueness

Ended

Description

You are given a string S. Your task is to write a program that comments if it has all u nique character or not (no repeated character).

If it has just unique character, output "Unique"

Else in all other cases, output "No"

Input

Input Format

First and the only line contains string S

Constraints

Length of S < 1000

Output

Output one string based on string

Sample Input 1

```
masai
```

Sample Output 1

```
No
```

```
function check(str){
    for(var i=0;i<str.length;i++){
        for(var j=i+1;j<str.length;j++){
            if(str[i]==str[j]){
                return false
            }



        }


    }
    return true
```

```
}

function runProgram(input){
 input=input.trim()
 var str=input
 //console.log(str)
var res=check(str)
if(res==true){
    console.log("Unique")
}
else{
    console.log("No")
}
}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
```

```
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Motu to Potlu house

Ended

Description

Motu decided to visit his friend Potlu. It turned out that the Motu's house is located at point 0 and his friend's house is located at point x (x > 0) of the coordinate line. In one step the Motu can move 1, 2, 3, 4 or 5 positions forward.

Determine, what is the minimum number of steps he need to make in order to get to his friend 's house.

Input

Input Format :

First and the only line contain the integer n which denotes the position of his friend's house.

Constraints :

1 <= n <= 10^6

Output

Output contains a single line denoting the minimum number of steps.

Sample Input 1

```
26
```

Sample Output 1

```
6
```

Hint

Output Explanation :

For n = 26, Motu can move as

5 --> 5 --> 5 --> 5 --> 5 --> 1

Hence he needed 6 steps to reach at position 26.

```
function runProgram(input){
    var N =+input
    check(N)
}
function check(N){
    var r = N%5
    var NewN = Math.floor(N/5)
    if(r==0){
```

```
        console.log(NewN)
    }
    else if(r>0){
        console.log(NewN+1)
    }
}
```

```
if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Distinct Substrings

Ended

Description

Given a string s of length n, find out the number of distinct substrings possible from t he given string.

Input

The first line of the input contains one integer t (1 ≤ t ≤ 100) — the number of test ca ses. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100) — the length of the string.

The second line of each test case contains a string s of length n.

Output

For each test case, print the answer: The number of distinct substrings possible from given string.

Sample Input 1

```
2
5
abcde
3
aaa
```

Sample Output 1

```
15
3
```

NEED TO BE DONE

https://oj.masaischool.com/contest/3720/problem/06

Hostel & Warden

Ended

Description

It's party time in a hostel, and you are partying with your friends in the hostel's lobby. You can visualise the lobby as a 1-dimensional line (x-axis). All of you are scattered along that X-axis. Suddenly, you come to know that the warden is coming to the host el. So, all of you want to hide inside a room. So, you will be given x-coordinates of ho stel rooms as well as your current location (as x-axis coordinate). Anyone can stay at his position, move one step right from x to x + 1, or move one step left from x to x -1. Any of these moves consume1 minute. You have to find out the minimum time in whi ch everyone in the lobby can go to any room and hide inside.

Input

Input Format

First line of input contains N whichis the total number of students present in the lobby

Second line contains N space-separatedintegers representing current x-coordinates of students scattered in lobby

The third line also contains N spaceseparated integers representing x-coordinates of the rooms present in the lobby

Constraints

```
N <= 1000000

-1000 <= x[i] <= 1000
```

Output

Print the minimum time as per the given condition in question

Sample Input 1

```
3

4 -4 2

4 0 5
```

Sample Output 1

```
4
```

Hint

Sample 1 Explanation

Assign student at position x = 4 to room at position x = 4 : Time taken is 0 minutes

Assign student at position x=-4 to room at position x = 0 : Time taken is 4 minutes

Assign student at position x=2 to room at position x = 5: Time taken is 3 minutes

After 4 minutes all of the students are in the rooms. Since, there is no combination p ossible where the last student's time is less than 4,

So, answer = 4.

Language:
JavaScript

```
function check(n,stud,room){
    let max=-Infinity
    for(let i=0;i<n;i++){
        if(stud[i]!=room[i]){
            let count=Math.abs(room[i]-stud[i])
            if(count>max){
                max=count
            }
        }
    }
    console.log(max)
}
```

```
function runProgram(input){
```

```
 input=input.trim().split("\n")
 let n=+input[0]
 let stud=input[1].trim().split(" ").map(Number);
 let room=input[2].trim().split(" ").map(Number);
 stud.sort((a,b)=>a-b)
 room.sort((a,b)=>a-b)
 check(n,stud,room)

}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {
```

```
  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Angry People

Ended

Description

You have a circular dining table and N angry people.You need to place them on the table.Each person has a anger value.

You can place them in any order on the circular table.Once you place them you need to calculate the Danger value of your arrangement.

The danger value of the arrangement is the maximum difference of anger values of all the adjacent seated persons.You need to keep this danger value as low as possible.

Input

Input Format

The first line contains N.

The second line contains the anger values of N persons.

Constraints

3<=N<=1000

1<=Hunger Value<=1000

Output

Output format

Print the minimum possible danger value.

Sample Input 1

```
4

5 10 6 8
```

Sample Output 1

```
4
```

```
function check(arr,n){
  arr.sort(function(a,b){return a-b});
  var out=[];
  for(var i=n-1; i>=0; i--){
      if(i%2===0){
          out.push(arr[i]);
      }
  }
  for( i=0; i<n; i++){
      if(i%2!==0){
          out.push(arr[i]);
      }
```

```
  }
   var j=0;
   var k=1;
  var max=-Infinity;
   while(j<n && k<n){
       minus=(Math.abs(out[j]-out[k]));
       j++;
       k++;
       if(minus>max){
           max=minus;
       }
   }
   console.log(max);
}
function runProgram(input){
    input=input.trim().split("\n");
    var n=+input[0];
    var arr=input[1].split(" ").map(Number);
    check(arr,n);
}
if (process.env.USER === "") {
  runProgram(``);

} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
```

```
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Solve Age of empires

Ended

Description

You are playing the popular game of "Age of Empires". You are the king of the empir e where you have 2N workers.

All workers will be grouped in the association with size 2, so a total of N associations have to be formed.

The building speed of the i-th worker is A[i].

To make an association, you pick up 2 workers. Let the minimum building speed bet ween both workers be x, then the association has the resultant building speed x.

You have to print the maximum value possible of the sum of building speeds of N as sociations if you make the associations optimally.

Input

First line contains an integer N, representing the number of associations to be made.

Next line contains 2N space separated integers, denoting the building speeds of 2N workers.

N <= 50000

A[i] <= 10000

Output

Print the maximum value possible of the sum of building speeds of all the association s.

Sample Input 1

```
2

1 3 1 2
```

Sample Output 1

```
3
```

Hint

Sample1Explanation

If you make an association using the first and third worker, and another using the sec ond and fourth worker, each association will have 1 and 2 resultant building speed, w hich has a total of 3.

```
function check(arr,N){


    function check(a,b){
      if(a>b){
          return b;
      }
      else if(b>a){
          return a
      }
      else{
          return a
      }
    }
  arr.sort((a,b)=>a-b);

  var i=0;
  var sum=0;
  var j=i+1;
```

```
    while(i<arr.length&&j<arr.length){

      sum=sum+check(arr[i],arr[j]);
      i=i+2;
      j=j+2;
    }
  console.log(sum)
}

function runProgram(input){
 input=input.trim().split("\n");
    var N=+input[0];
    var arr=input[1].trim().split(" ").map(Number);
    check(arr,N)

}




if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";
```

```
process.stdin.on("data", function (input)
{
    read += input;

});

process.stdin.on("end", function () {

  read = read.replace(/\n$/, "");

read = read.replace(/\n$/, "");

  runProgram(read);
  });

 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Opp Quick Sort

Ended

Description

Given a list of n integers. Write a program for quick sort algorithm such that it reverse s the list in descending order. You must not write any other sorting algorithm

Input

First line contains n which is the number of elements present in the array

Second line contains n space-separated integers

Output

Output the elements present in the array sorted in descending order

Sample Input 1

```
5

2 3 1 4 5
```

Sample Output 1

```
5 4 3 2 1
```

NEED TO DONE

Gas Station

Ended

Description

There are n gas stations along a circular route, where the amount of gas at the ith st ation is gas[i].

You have a car with an unlimited gas tank and it costs cost[i]of gas to travel from the ith tation to its next (i + 1)th station. You begin the journey with an empty tank at one of the gas stations.

Given two integer arrays gas and cost, find out the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise print-1. If ther e exists a solution, it is guaranteed to be unique

Input

Input Format

The first line contains the number of gas station N

The second line contains the values of gas[i]

The third line contains the values of cost[i]

Constraints

1<=N<=5*10^5

1<= gas[i] <=1000

1<= cost[i] <=1000

Output

Printthe starting gas station's index if you can travel around the circuit once in the clo ckwise direction, otherwise print `-1`

Sample Input 1

```
5

1 2 3 4 5

3 4 5 1 2
```

Sample Output 1

```
3
```

Hint

Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = 0 + 4 = 4
Travel to station 4. Your tank = 4 - 1 + 5 = 8

Travel to station 0. Your tank = 8 - 2 + 1 = 7

Travel to station 1. Your tank = 7 - 3 + 2 = 6

Travel to station 2. Your tank = 6 - 4 + 3 = 5

Travel to station 3. The cost is 5.

Your gas is just enough to travel back to station 3.Therefore, return 3 as the starting i ndex.

```
function check(n,gas,cost){
    let current=0;
    let total=0;
    let start=0;
    for(let i=0;i<n;i++){
        let rem=gas[i]-cost[i];
        if(current<0){
            start=i;
            current=rem;
        }
        else{
            current+=rem;
        }
        total+=rem;
    }
    if(total<0){
        return -1;
    }
    else{
        return start;
    }
}
```

```
function runProgram(input){
 input=input.trim().split("\n");
 var n=+input[0]
 var gas=input[1].trim().split(" ").map(Number)
 var cost=input[2].trim().split(" ").map(Number)
 console.log(check(n,gas,cost))
```

```
}
```

```
if (process.env.USER === "")
 {

runProgram(``);

 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });
```

```
 process.on("SIGINT", function () {

   read = read.replace(/\n$/, "");

runProgram(read);

process.exit(0);

});

}
```

Bon Appetit

Ended

Description

Abhishek and Akash were two friends and are deciding to split the bill of the food they eat for dinner. Each of them will only pay the bill for the items they consume. Abhishek gets the check and calculates Akash's portion. You must determine if his calculation is correct.

For example, assume the bill has the following prices bill = [2,4,6]. Akash declines to eat item k = bill[2] which costs 6. If Abhishek calculates the bill correctly, Akash will pay (2+4)/2 = 3. If he includes the cost of the bill[2], he will calculate (2+4+6)/2 = 6. In the second case, he should refund to Akash.

Input

Input Format

First-line contains T, no of test cases.

The first line of each test case contains two space-separated integers n and k, the number of items ordered, and the 0-based index of the item that Akash did not eat.

The second line of each test case contains space-separated integers bill[i] where 0 <= i < n.

The third line of each test case contains an integer, b, the amount of money that Abh ishek charged Akash for his share of the bill.

Constraints

1 <= T <= 10

2 <= n <= 10^5

0 <= k < n

0 <= bill[i] <= 10^4

0 <= b <= Sum of all elements of bill array.

Output

For each test case, If Abhishek did not overcharge Akash, print Bon Appetit on a ne w line; otherwise, print the difference (i.e., chargedAmount - actualAmount) that Abhi shek must refund to Akash. This will always be an integer.

Sample Input 1

```
2

4 1

3 10 2 9

12

4 1

3 10 2 9

7
```

Sample Output 1

```
5

Bon Appetit
```

Hint

For the first test case,

Akash didn't eat item bill[1] = 10, but he shared the rest of the items with Abhishek. T he total cost of the shared items is 3+10+2+9 = 14 and, split in half, the cost per pers on is actualAmount = 7. Abhishek charged him chargedAmount = 12 for his portion o f the bill. We print the amount Akash was overcharged, 5, on a new line.

For the second test case,

Akash didn't eat item bill[1] = 10, but he shared the rest of the items with Abhishek. T he total cost of the shared items is 3+10+2+9 = 14 and, split in half, the cost per pers on is actualAmount = 7. Abhishek charged him chargedAmount = 7 for his portion of the bill. We print the Bon Appetit as the Abhishek does not overcharge Akash, on a n ew line.

```
function check(n,m,arr,ans){

    var sum=0;
    for(var k=0;k<n;k++)
    {
        if(k!=m)
        {
            sum=sum+arr[k]
        }
    }
    sum=Math.floor(sum/2)
    if(ans>sum)
    {
        return ans-sum
    }
    else
    {
        return "Bon Appetit"
    }
}
```

```
function runProgram(input) {
   input=input.split("\n");
   var tc=+input[0];
   var line=1;

   for(var i=0;i<tc;i++){

       var [n,m]=input[line].split(" ").map(Number);
       line++;
       var arr=input[line].split(" ").map(Number);
       line++;
       var ans=+input[line]
       line++;
    //   console.log(n,m,arr,ans);
      console.log(check(n,m,arr,ans))
   }
 }
 if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
```

```
    process.exit(0);
  });
}
```

https://oj.masaischool.com/contest/3747/problems

Array Parts of Four

Ended

Description

- You are given an array stored in a variable with the name `arr`
- The size of the array is stored in a variable with the name `N`
- The value stored in `N` is always divisible by 4
- You have to find the value of the equation `-2*c1 + c2 + 2*c3 + c4`, such that

```
The array is divided into 4 halves

c1 -> The sum of the first half of the array

c2 -> The sum of the second half of the array

c3 -> The sum of third half of the array

c4 -> The sum of the fourth half of the array
```

- For example, consider the value stored in `N = 8`, and `arr = [1 2 3 4 5 6 7 8]`
- Now, `c1 = 1 + 2 = 3`, `c2 = 3 + 4 = 7`, `c3 = 5 + 6 = 11`, `c4 = 7 + 8 = 15`
- Therefore, the value of the equation `2*(3) + 7 + 2*(11) + 15 = 6 + 7 + 22 + 15 = 50`, which is the required output

Input

- The first line of the input contains the value stored in `N`
- The next line contains the value stored in `arr`

Output

- Print the value of the equation as explained in the problem statement

Sample Input 1

```
8

1 2 3 4 5 6 7 8
```

Sample Output 1

```
50
```

```
function check(N,arr){
 var c1=0,c2=0,c3=0,c4=0;
 var sum=0;
 var div=N/4;
 var l1=0;
 var l2=div
for(var i=l1;i<l2;i++){
    c1=c1+arr[i]
}
l1=l1+div
l2=l2+div
 for(var i=l1;i<l2;i++){
    c2=c2+arr[i]
}
l1=l1+div
l2=l2+div
 for(var i=l1;i<l2;i++){
    c3=c3+arr[i]
}
l1=l1+div
l2=l2+div
 for(var i=l1;i<l2;i++){
    c4=c4+arr[i]
}
var res=2*(c1)+c2+2*(c3)+c4;
console.log(res)
```

}

Generate Permutations

Ended

Description

Given a collection of numbers, return all possible permutations.

NOTE:

No two entries in the permutation sequence should be the same.

For the purpose of this problem, assume that all the numbers in the collection are uni que.

USING BUILT-IN LIBRARY FUNCTION TO PERMUTE WILL LEAD TO DISQUALI FICATION

Input

Input Format :

The first line of input contain an integer n - denoting the size of array

The next line contain n distinct integers, A[1],A[2]...A[N]

```
var P = [];
function p(arr,i,curr_arr){
    if(i>= arr.length){
        P.push(curr_arr.join(" "));
        return;
    }
    for(var j=0; j<arr.length; j++){
        if(!curr_arr.includes(arr[j])){
            curr_arr.push(arr[j]);
            p(arr,i+1,curr_arr);
            curr_arr.pop();
        }
```

```
    }
}
function runProgram(input) {
    input = input.trim().split("\n");
    var n = parseInt(input[0]);
    var arr  = input[1].trim().split(" ").map(Number);
    p(arr,0,[]);

    for(let i=0; i<P.length; i++){
        console.log(P[i]);
    }
}
process.stdin.resume();
process.stdin.setEncoding("ascii");
let read = "";
process.stdin.on("data", function (input) {
    read += input;
});
process.stdin.on("end", function () {
    read = read.replace(/\n$/, "")
    runProgram(read);
});
process.on("SIGINT", function () {
    read = read.replace(/\n$/, "")
    runProgram(read);
    process.exit(0);
});
```

https://oj.masaischool.com/contest/3775/problems

Jerry and Cheese

Ended

Description

Jerry is located at the origin (0,0) and he loves cheese. There are n stores in the locality with each having it's x and y coordinates.

Jerry wants to go to the nearest store from his location and buy cheese from there. Jerry can move diagonally (x ± 1, y ± 1), horizontally (x ± 1, y) and vertically (x, y ± 1) in one move. He wants to find out the number of shops available which are nearest to him.

Jerry is busy playing with Tom, so can you please help him out?

Input

The first line of the input contains one integer t (1 ≤ t ≤ 10) — the number of test cases. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100000) — the number of shops in the neighborhood.

The next n lines of each test case contain 2 integers xi and yi (1 ≤ xi, yi ≤ 100) - the coordinates of each shop.

Output

For each test case, print the answer: The number of shops which are nearest to Jerry.

Sample Input 1

```
1
5
2 2
3 2
4 2
2 1
0 2
```

Sample Output 1

```
3
```

NEED TO BE DONE

Is Even Sum Possible?

Ended

Description

Given an array a which contains n integers in it. Find out if there exists a subset of th e array such that the sum of its elements is even.

Input

The first line of the input contains one integer t (1 ≤ t ≤ 10) — the number of test cas es. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100000) — the len gth of the array a.

The second line contains n integers a1,a2,…,an (1 ≤ ai ≤ 10000), elements of the arr ay a.

Output

For each test case, print the answer: "Yes" if possible or else "No" if not possible (wit hout quotes).

Sample Input 1

```
2

1

5

4

1 2 3 4
```

Sample Output 1

```
No

Yes
```

```
function check(arr, n)
{
    var s1 = 0;
    var s2 = 0;
    if(n===1 && arr[0]%2!==0)
    {
        return "No";
    }
    for(var i=0; i<n; i++)
    {
        if(s1%2===0 || s2%2===0)
        {
            return "Yes";
        }
        else
        {
            return "No";
        }
        if(arr[i]%2===0)
        {
         s1++;
        }
        else
        {
            s2++  ;
        }
    }
}

function runProgram(input)
{
    input=input.trim().split("\n");
    var tc=+input[0];
    var line=1;
    for(var i=0; i<tc; i++)
```

```
    {
        var n=+input[line];
        line++;
        var arr=input[line++].trim().split(" ").map(Number);
        console.log(check(arr, n));
    }
}

if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Local Maxima

Ended

Description

Given an array of n elements, find out the number of elements in the array which are local maxima.

A local maxima is defined as an element that is greater than it's immediate neighbors (Do not consider the first and the last element since they have only one neighbor).

For example, an ith element has (i+1) and (i-1) as it's immediate neighbors.

Also, if the number of the elements is less than 3, then print -1 since it's not possible.

Input

The first line of the input contains one integer t (1 ≤ t ≤ 10) — the number of test cas es. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100000) — the nu mber of mountains peaks in the valley.

The second line contains n integers (1 ≤ ai ≤ 1000) - the height of each mountain pe ak.

Output

For each test case, print the answer: The number of local maxima if n > 2 or else -1.

Sample Input 1

```
2

2

1 2

10

884 729 768 201 266 494 597 328 705 287
```

Sample Output 1

```
-1

3
```

```
function check(arr,n)
{
    var count = 0;
```

```
    for(var i = 0; i <= n; i++)
    {
    if(arr.length < 3)
    {
       return "-1"
    }
    else
        if(arr[i] > arr[i+1] && arr[i] > arr[i-1])
        {
            count++
        }
    }
    return count
}

function runProgram(input)
{
    input=input.trim().split("\n");
    var tc=+input[0];
    var line=1;
    for(var i=0; i<tc; i++)
    {
        var n=+input[line];
        line++;
        var arr=input[line].trim().split(" ").map(Number);
        line++;
        console.log(check(arr,n));
    }
}

if (process.env.USER === "") {
  runProgram(``);
} else {
  process.stdin.resume();
  process.stdin.setEncoding("ascii");
```

```
  let read = "";
  process.stdin.on("data", function (input) {
    read += input;
  });
  process.stdin.on("end", function () {
    read = read.replace(/\n$/, "");
    read = read.replace(/\n$/, "");
    runProgram(read);
  });
  process.on("SIGINT", function () {
    read = read.replace(/\n$/, "");
    runProgram(read);
    process.exit(0);
  });
}
```

Spirals and Diagonals

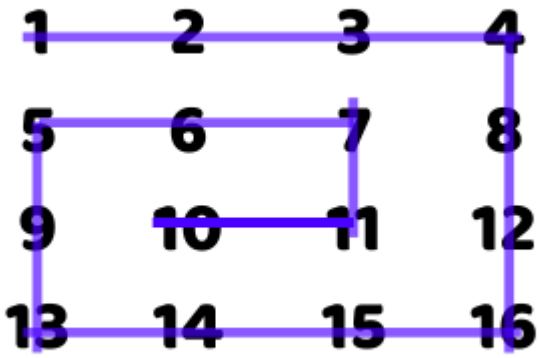
Ended

Description

You are given an integer n. The next line is an array A which contains n*n elements. The spiral traversal of a square matrix of dimension (n x n) results in the given array.

Calculate the sum of all elements which are on the diagonals of the square matrix.

The matrix traversal happens in the manner shown in the image below



Input

The first line of the input contains one integer t (1 ≤ t ≤ 10) — the number of test cas es. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100)

The second line contains n*n integers (1 ≤ Ai ≤ 1000) - The spiral traversal of a squa re matrix of dimension n x n.

Output

For each test case, print the answer: The sum of elements present in the diagonals o f the matrix.

Sample Input 1

```
3

1

1

2

1 2 3 4

3

1 2 3 4 5 6 7 8 9
```

Sample Output 1

```
1

10

25
```

Sample Input 2

```
1

4

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
```

Sample Output 2

```
68
```

NEED TO BE DONE

Faulty Direction

Ended

Description

Given a string of length n which consists of characters {L, R, U, D}.

Initially, it can be assumed that you are standing at origin (x=0 and y=0). Here, L me ans left (x-1), R means right (x+1), U means up (y+1) and D means Down (y-1).

You are given this string so that the directions can be followed accordingly to reach a destination. You came to know that there are some string that is faulty and following t hem ends you up to your initial position that is the origin.

You don't want to start the trip if the string provided is faulty.

Find whether the string is faulty or not.

For ease of the problem, do consider you are always facing the north direction.

Input

The first line of the input contains one integer t (1 ≤ t ≤ 100) — the number of test ca ses. Then t test cases follow.

The first line of each test case contains a single integer n (1 ≤ n ≤ 100) — the length of the string.

The second line of each test case contains a string s of length n.

Output

For each test case, print the answer: "Yes" if it returns to its initial position or else "N o" (without quotes).

Sample Input 1

```
2
5
RLRUD
4
LRUD
```

Sample Output 1

```
No
Yes
```

```
function check(n,str){
    var up=0;
    var down = 0;
    var left = 0, right = 0;
    for (let i = 0; i < n; i++)
        {

            if (str[i] == 'U')
                up++;
```

```
            else if (str[i] == 'D')
                down++;

            else if (str[i] == 'L')
                left++;

            else if (str[i] == 'R')
                right++;
        }
}


function runProgram(input){
 input=input.trim().split("\n")
 var tc=+input[0]
 var line=1;
 for(var i=0;i<tc;i++){
     var n=+input[line];
     line++;
     var str=input[line];
     line++;
     console.log(check(n,str))
 }

}




if (process.env.USER === "")
 {

runProgram(``);
```

```
 }
else
 {
   process.stdin.resume();

  process.stdin.setEncoding("ascii");

 let read = "";

 process.stdin.on("data", function (input)
 {
     read += input;

 });

 process.stdin.on("end", function () {

   read = read.replace(/\n$/, "");

 read = read.replace(/\n$/, "");

   runProgram(read);
   });

  process.on("SIGINT", function () {

    read = read.replace(/\n$/, "");

 runProgram(read);

 process.exit(0);

});

}
```

Rotate Linked List

Ended

Description

Given a linked list, rotate the list to the right by k places, where k isnon-negative.

Complete the function, and return the head of the updated list

Input

This is a function complete problem.

Refer to sample input for understanding

The first line contains the number of nodes (n)

Next n lines contains the nodes of the linked list

Next line contains k

n <= 1000

list[i] <= 1000

Output

Complete the function

Sample Input 1

```
3
1
2
3
2
```

Sample Output 1

```
2 3 1
```

```
const LinkedListNode = class {
    constructor(nodeData) {
        this.data = nodeData;
        this.next = null;
    }
};

var rotateRight = function(head, k) {
    if(head==null){
        return null}
    let curr=head;

    let len=1;
    while(curr.next!=null){
        curr=curr.next;
        len++;
    }
    curr.next=head
    let prelen=len-k%len-1
    var pre=head
    while(prelen>0){
        pre=pre.next;
        prelen--;
    }
    head=pre.next;
    pre.next=null;
    return head

};
```