C:\Program Files\MongoDB\Server\6.0\data\

1) Setwet's Fish Pond

https://my.newtonschool.co/playground/code/fzizuvcdgi9c/

Rahul's Groceries (Contest)

https://my.newtonschool.co/playground/code/klqvbykc88x9/7

Array construction

https://my.newtonschool.co/playground/code/o51g88snh5lp/

Too Close (Contest)

https://my.newtonschool.co/playground/code/ildvnfh8y3le/

Power of Three - Contest 15-01-23

import java.io.\*;

import java.lang.\*;

class Main

{

public static void main(String args[])throws IOException

{

InputStreamReader isr = new InputStreamReader(System.in);

BufferedReader br = new BufferedReader(isr);

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

while(t-- >0)

{

long n = Long.parseLong(br.readLine());

if(n == 0)

{

System.out.println(1);

continue;

}

System.out.println(powerOfThree(n));

}

}

public static long powerOfThree(long n)

{

if(n<=0)

return 0;

long p = (long)(Math.log(n)/Math.log(3));

if(gpSum(p)>=n)

return power(3,p) + powerOfThree(n - power(3,p));

else

return power(3,p+1);

}

public static long gpSum(long p)

{

return (power(3,p+1)-1)/2;

}

public static long power(long base, long p)

{

if(p==0)

return 1;

else

return(base\*power(base,p-1));

}

}

Buildings

https://my.newtonschool.co/playground/code/uj7affo0kcz1/

Sum Discrepancy (Contest

https://my.newtonschool.co/playground/code/o27h9mijnr5g/

Completely different

medium

https://my.newtonschool.co/playground/code/ninjg8hsl7bu/

Derrangement Exercise (Contest

https://my.newtonschool.co/playground/code/4sa8z4jf598u/

Sum Discrepancy (Contest)

https://my.newtonschool.co/playground/code/6yomrcnwvgti/

Morse Convertor

easy

https://my.newtonschool.co/playground/code/vv2d34um9fef/

Substring Reversal

easy

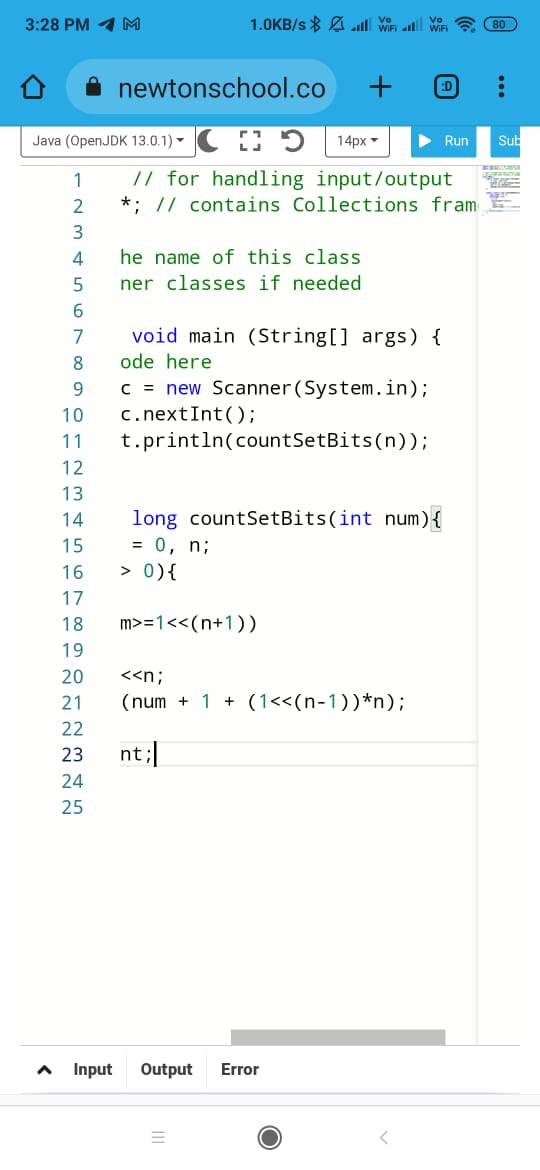
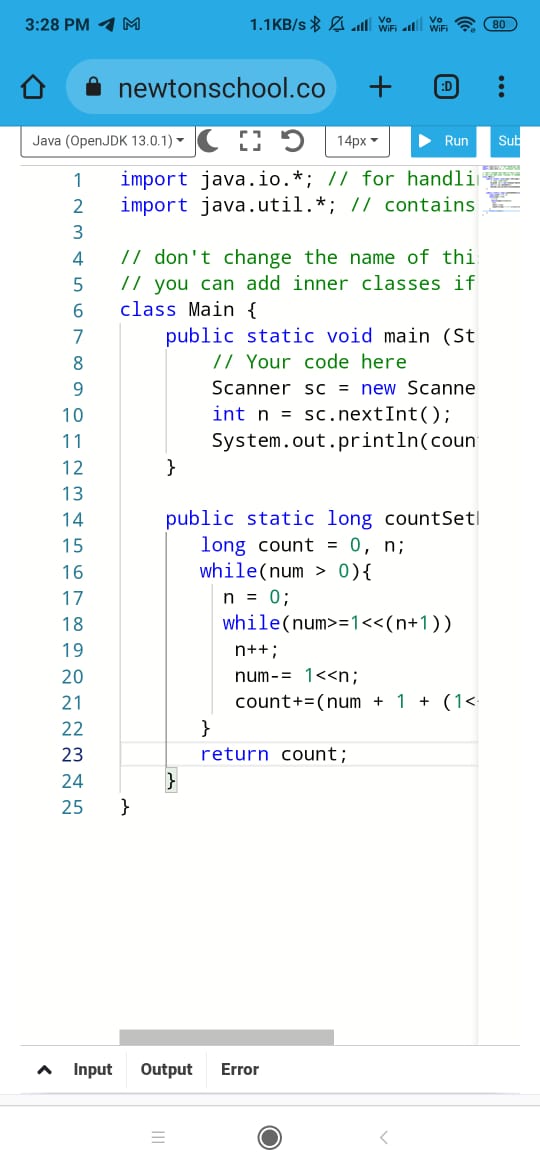
https://my.newtonschool.co/playground/code/halm8b5yfztd/

K closest points

easy

https://my.newtonschool.co/playground/code/yha4vfnkgt55/

Sara and Coins



Getting Memory Limit Exceeded error

Basics of Programming

**Me in middle**

**easy**

Time Limit: 2 sec  
Memory Limit: 128000 kB

**Problem Statement**

You are given an array of size n containing distinct integers, where n is odd. Find the element which has the same number of lesser elements and the same number of greater elements in the array.

**Input**

First line of the input contains an integer, N, which denotes the length of the array. Next N inputs are elements of the array.  
  
Constraints  
1 <= N <= 20  
-10000 <= Arr[i] <= 10000

**Output**

Single integer n which has the same number of lesser elements and the same number of greater elements.

**Example**

Input:  
3  
3 1 2  
  
Output:  
2  
  
Explanation:-  
2 has one greater element 3 and one smaller element 1  
  
Input:  
5import java.io.\*; // for handling input/output

import java.util.\*; // contains Collections framework

// don't change the name of this class

// you can add inner classes if needed

public class Main {

    ststic int findElement(int arr[], int n){

        HashMap<Integer,Integer>hm = new HashMap<>();

        int count = 0;

        for (int i = 0;i<n;i++){

            if (hm.containsKey(arr[i]))

            hm.put(arr[i],

            hum.get(arr[i])+1);

            else

            hm.put(arr[i],1);

        }

        for(int i =0;i< n;i++){

            count = 0;

            for(int j =0;j<i;i++){

                if(arr[j]<arr[i])

                count++;

            }

            for(int j = i + 1;j<n;j++){

                if (arr[j]<arr[i])

                count++;

            }

            if(count == hm.get(arr[i]))

            return arr[i];

        }

        return-1;

    }

    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        System.out.print("enter the ELEMENTS OF Array:");

        for(int i = 0;i<n;i++){

        arr[i] = sc.nextInt();

        }

        int res = findElement(arr,n);

        if(res!=-1)

        System.out.print("the element of with same number of"+ "lass and greater elements is"+res);

        else

        System.out.print("no such element exist");

    }

    }

2 3 4 9 1  
  
Output:  
3