PROBLEM SOLVING DAY 10

Q1

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'gameWithCells' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER n
     *  2. INTEGER m
     */

    public static int gameWithCells(int n, int m) {
      if(m%2==0&&n%2==0)
      return (m*n)/4;
      else if(m%2==0&&n%2!=0)
      return (m*(n-1)/4+(m)/2);
      else if(m%2!=0&&n%2==0)
      return (n*(m-1)/4+(n)/2);
      else
```

```
        return (((m-1)*(n-1))/4+(m+n)/2);
    // Write your code here


    }


}


public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));


        String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");


        int n = Integer.parseInt(firstMultipleInput[0]);


        int m = Integer.parseInt(firstMultipleInput[1]);


        int result = Result.gameWithCells(n, m);


        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();


        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

PROBLEM SOLVING DAY 10

```
Q2      import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'connectingTowns' function below.
     *
     * The function is expected to return an INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER n
     *  2. INTEGER_ARRAY routes
     */

    public static int connectingTowns(int n, List<Integer> routes) {
    // Write your code here
    long s=1,a=1,c=1234567;
    for(int i=0;i<n-1;i++){

        s=routes.get(i);
        a=a*s;
        if(a>1234567){
            a %= 1234567;
        }
    }
        return (int)a ;
    }

}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        int t = Integer.parseInt(bufferedReader.readLine().trim());

        IntStream.range(0, t).forEach(tItr -> {
            try {
                int n = Integer.parseInt(bufferedReader.readLine().trim());

                List<Integer> routes = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))
                    .map(Integer::parseInt)
                    .collect(toList());

                int result = Result.connectingTowns(n, routes);

                bufferedWriter.write(String.valueOf(result));
                bufferedWriter.newLine();
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        });

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

Q3

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    /*
     * Complete the 'solve' function below.
     *
     * The function is expected to return a LONG_INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER n
     *  2. INTEGER m
     */
    public static long solve(int n, int m) {
        return (long)m*n-1;
    // Write your code here

    }

}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");

        int n = Integer.parseInt(firstMultipleInput[0]);

        int m = Integer.parseInt(firstMultipleInput[1]);

        long result = Result.solve(n, m);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

PROBLEM SOLVING DAY 10

```
Q4 import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        int n = Integer.parseInt(bufferedReader.readLine().trim());
        int max_sum=1,max_number=1;
        for(int i=2;i<=n;i++){
            if(n%i==0){
                int a=i,s=0;
                while(a>0){
                    s=s+a%10;
                    a=a/10;
                }
                if(s>max_sum){
                    max_sum=s;
                    max_number=i;
                }
            }
        }
        System.out.println(max_number);
        bufferedReader.close();
    }
}
```

```
Q5 import java.util.Scanner;
public class D10Q5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        long s = 0;
        long t = sc.nextLong();
        for (long j = 1; j <= t; j++) {
            long l = sc.nextLong();
            long b = sc.nextLong();
            if (l == b)
                s = 1;
            else {
                if (l < b) {
                    for (long i = 1; i <= l; i++) {
                        if ((l % i == 0) && (b % i == 0))
                            s = i;
                    }
                } else {
                    for (long i = 1; i <= b; i++) {
                        if ((l % i == 0) && (b % i == 0))
                            s = i;
                    }
                }
            }
            s=(l*b)/(s*s);
            if(l==b)
            System.out.println("1");
            else
            System.out.println(s);
        }
    }
}
```

```
Q6 import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



public class Solution {
    public static void main(String[] args) throws IOException {
        Scanner sc=new Scanner(System.in);
        int t=sc.nextInt();
        while(t-->0){
            long n=sc.nextLong();
            long k=sc.nextLong();
            long temp=0,count=1;
            long a[]=new long[(int)n];
            long b[]=new long[(int)n];
            for(long i=0;i<n;i++)
            a[(int)i]=i;
            for(long i=0;i<n;i++){
                if(i%2==0){
                     b[(int)i]=a[(int)(n-count)];
                     count++;
                }
                else
                b[(int)i]=a[(int)(count-2)];
            }
            for(long i=0;i<n;i++){
                if(b[(int)i]==k)
                count=i;
            }
            System.out.println(count);
        }
    }
}
```

```
Q7 import java.util.Scanner;
import java.util.Scanner.*;
public class D10Q7 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        long r=sc.nextLong();
        long c=sc.nextLong();
        long z=0;
        if(r==1)
            z=(c-1)*2;
        else if(r==2)
            z=(c-1)*2+1;
        else{
            if(r%2!=0)
                z=(r-1)*5+(c-1)*2;
            else
                z=(r-2)*5+1+(c-1)*2;
        }
        System.out.println(z);
    }
}
```

```
Q 8 import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

class Result {

    /*
     * Complete the 'halloweenParty' function below.
     *
     * The function is expected to return a LONG_INTEGER.
     * The function accepts INTEGER k as parameter.
     */
    public static long halloweenParty(int k) {


    long z=k;

    if(z%2!=0)

    z=(z/2)*(z/2+1);

    else

    z=(z/2)*(z/2);

    return z;


    }


}


public class Solution {

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

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

        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));


        int t = Integer.parseInt(bufferedReader.readLine().trim());


        for (int tItr = 0; tItr < t; tItr++) {

            int k = Integer.parseInt(bufferedReader.readLine().trim());


            long result = Result.halloweenParty(k);


            bufferedWriter.write(String.valueOf(result));

            bufferedWriter.newLine();

        }


        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

Q 109

```
    import java.util.*;
public class D10Q9 {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        long n=sc.nextLong();
        long m=sc.nextLong();
        long avg=0;
        while(m>0){
            long a=sc.nextLong();
            long b=sc.nextLong();
            long k=sc.nextLong();
            avg=avg+(b-a+1)*k;
            m--;
        }
        System.out.println(avg/n);
    }
}
```