Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	0	100	0	400

Task A ()

```
#include <iostream>
#include "algorithm"
#include "vector"
#include "set"
#include "map"

typedef unsigned long long ull;
typedef long ll;
#define fe(a,b) for(auto a: b)
#define forAll(x) x.begin(), x.end()
#define fp(i,s,f) for(auto i = s; i!=f; i++)
#define f(i,s,f) for(int i = s; i<f; i++)
#define add(x) push_back(x)
#define magic_words ios_base::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr);

using namespace std;
int main() {
    magic_words
    int n;
    cin >> n;
    ll now = n/2;
    while(now!=(now+n)/2) now=(now+n)/2;
    cout << now;

    return 0;
}
```

Task B ()

```cpp
#include <iostream>
#include "algorithm"
#include "vector"
#include "set"
#include "map"
#include "math.h"

typedef unsigned long long ull;
typedef long ll;
#define fe(a,b) for(auto a: b)
#define forAll(x) x.begin(), x.end()
#define fp(i,s,f) for(auto i = s; i!=f; i++)
#define f(i,s,f) for(int i = s; i<f; i++)
#define add(x) push_back(x)
#define magic_words ios_base::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr);

using namespace std;

double distance(pair<double, double> f, pair<double, double> s){
    return sqrt((f.first-s.first)*(f.first-s.first)+(f.second-s.second)*(f.second-s.second));
}
int main() {
    magic_words
    int n;
    cin >> n;
    if(n==6){
        pair<double, double> points[6];
        double sumx =0 , sumy=0;
        f(i, 0, 6){
            double x,y;
            cin >> x >> y;
            sumx+=x;
            sumy+=y;
            points[i] = make_pair(x,y);
        }
        sumx/=6;
        sumy/=6;
        pair<double, double> center = make_pair(sumx, sumy);
        int count = 0;
        pair<double, double> choosen[3];
        f(i,0,6){
            bool can = true;
            f(j, 0, count){
                if(j==i) continue;
                if(abs(distance(choosen[j], points[i])-distance(points[i], center))<=0.5){
                    can = false;
                    break;
                }
                if (abs(2*center.first-points[i].first-choosen[j].first)<=0.5 && abs(2*center.
                    second-points[i].second-choosen[j].second)<=0.5){
                    can = false;
                    break;
                }
                if(count==3){
                    break;
                }
            }
            if(can){
                choosen[count] = points[i];
                count++;
            }
        }
        for(auto p: choosen) cout << p.first << '␣' << p.second << endl;
    }
    else{
        pair<double, double> points[3];
        f(i,0,3) {
            double x,y;
            cin >> x >> y;
            points[i] = make_pair(x,y);
        }
        pair <double, double> center;
        double sumx=0, sumy=0;
        f(i, 0, 3){
            sumx+=points[i].first;
            sumy+=points[i].second;
        }
        sumx/=3;
        sumy/=3;
        center = make_pair(sumx, sumy);

        cout << points[0].first << '␣' << points[0].second << endl;
        cout << 2*center.first-points[2].first << '␣' << 2*center.second-points[2].second << endl;
        cout << points[1].first << '␣' << points[1]. second << endl;
        cout << 2*center.first-points[0].first << '␣' << 2*center.second-points[0].second << endl;
        cout << points[2].first << '␣' << points[2].second << endl;
        cout << 2*center.first-points[1].first << '␣' << 2*center.second-points[1].second << endl;
    }

    return 0;
}
```

Task C ()

```java
import org.w3c.dom.ls.LSOutput;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;


public class MainClass {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String correct = scanner.next();
        int n = scanner.nextInt();
        String[] answers = new String[n];
        for (int i = 0; i < n; i++)
            answers[i] = scanner.next();
        long sum = 0;
        for (int i = 0; i < n; i++) {
            int dp[][] = new int[answers[i].length()][correct.length()];
            for (int[] d : dp) Arrays.fill(d, 0);
            int max = 0;
            for (int j = 0; j < answers[i].length(); j++) {
                for (int k = 0; k < correct.length(); k++) {
                    if (answers[i].charAt(j) == correct.charAt(k)) {
                        int prevMax = 0;
                        if (j != 0)
                            for (int l = 0; l < k; l++) prevMax = Math.max(prevMax, dp[j - 1][l]);
                        dp[j][k] = prevMax + 1;
                        max = Math.max(prevMax + 1, max);
                    }
                }
            }

            for (int b : dp[answers[i].length() - 1]) max = Math.max(max, b);
            sum += correct.length() - max;
        }
        System.out.println(sum);
    }
}
```

Task D ()

Task E ()

```java
import org.w3c.dom.ls.LSOutput;

import java.awt.*;
import java.util.*;

class Point{
    long x, y;
    Point(long x, long y){
        this.x = x;
        this.y = y;
    }
}

public class MainClass {
    static long maxX, maxY;
    static HashSet<Long> set;
    static ArrayList<Point> queue;
    static Point getNextPolong(long a, long c){
        return new Point(a*maxX+queue.get((int) c).x, queue.get((int) c).y);
    }
    static Point q(Point p1, Point p2, Scanner scanner){
        System.out.println("?␣" + p1.y + "␣" + p1.x + "␣" + p2.y + "␣" + p2.x);
        System.out.flush();
        long a = scanner.nextLong();
        long b = scanner.nextLong();
        return new Point(b, a);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        maxY = scanner.nextLong();
        maxX = scanner.nextLong();
        long n = scanner.nextLong();
        queue = new ArrayList<>();
        for (long i = 0; i < n; i++) {
            long y = scanner.nextLong();
            long x = scanner.nextLong();
            queue.add(new Point(x, y));
        }
        set = new HashSet<Long>();
        long squers = (long) Math.pow(2.0, n);
        for (long i = 0; i < squers; i++) {
            set.add(i);
        }
        for (long i = 0; i < n; i++) {
            Iterator<Long> iterator = set.iterator();
            HashSet<Long> toRemove = new HashSet<>();
            long wasSize = set.size();
            while(iterator.hasNext()){
                long a = iterator.next();
                long b = iterator.next();
                toRemove.add((q(getNextPolong(a, i), getNextPolong(b, i), scanner).x-1)/maxX);
            }
            set.removeAll(toRemove);
            long nowSize = set.size();
            Iterator<Long> iter = set.iterator();
            HashSet<Long> newSet = new HashSet<>();
            for (long j = 0; j < wasSize/2; j++) newSet.add(iter.next());
            set=newSet;
        }
        System.out.println("!␣1␣" + (set.iterator().next()*maxX+1));
        System.out.flush();
    }
}
```

Task F ()

```
#include <iostream>
#include "algorithm"
#include "vector"
#include "set"
#include "map"
#include "math.h"

typedef unsigned long long ull;
typedef long ll;
#define fe(a,b) for(auto a: b)
#define forAll(x) x.begin(), x.end()
#define fp(i,s,f) for(auto i = s; i!=f; i++)
#define f(i,s,f) for(int i = s; i<f; i++)
#define add(x) push_back(x)
#define magic_words ios_base::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr);

using namespace std;


int main() {
    ull mod = 1000000007;
    ull n, m;

    cin >> n >> m;
    if(n==2){
        cout << 1;
        return 0;
    }
    for (ull i = 1; i < (n-1)*(n-1); ++i) {
        cout << 0 << '␣';
    }
    ull ans = 1;
    for (ull j = n; j > 2; --j) {
        ans*=j;
        ans%=mod;
        cout << ans << '␣';
    }
    return 0;
}
```