KANTON SARAJEVO

MINISTARSTVO ZA ODGOJ I OBRAZOVANJE
KANTONA SARAJEVO

Kantonalno takmičenje iz informatike za srednje škole

05. April 2023. GODINE

Rezultati

Poredak takmičara

Mjesto	Ime i prezime	Škola	Fabrika	Raketne čizme	Biljke	Profesor Srmen	Zapetljanost	Ukupno
1	Benjamin Mujkić	JU Druga gimnazija	100	100	100	100	100	500
2	Faruk Ibrahimović	JU Druga gimnazija	100	100	100	46	100	446
3	Admir Zatega	JU Druga gimnazija	100	100	100	100	35	435
4	Farah Demirović	PU Richmond Park International Secondary School	100	100	100	21	6	327
5	Faruk Hodžić	JU Prva gimnazija Sarajevo	100	100	42	21	17	280
6	Bakir Činjarević	Prva bošnjačka gimnazija	100	90	79	0	6	275
7	Faruk Demirović	PU Richmond Park International Secondary School	100	100	43	8	0	251

8	Emina Hasanbegović	JU Druga gimnazija	100	100	24	0	17	241
9	Faris Čišija	JU Treća gimnazija	100	100	0	21	17	238
10	Emira Ibrahimović	JU Druga gimnazija	100	100	11	8	6	225
11	Muhamed Avdić	JU Četvrta gimnazija Ilidža	100	100	0	0	17	217
12	Nadir Hrustanbegović	JU Treća gimnazija	100	100	0	0	17	217
13	Appa Bugis Mubarak	Srednja elektrotehnička škola Sarajevo	100	100	0	0	0	200
14	Amer Bećarević	JU Četvrta gimnazija Ilidža	100	80	0	0	17	197
15	Nidal Vatreš	PU Richmond Park International Secondary School	100	80	0	0	17	197
16	Adna Mujić	JU Prva gimnazija Sarajevo	100	90	0	0	0	190
17	Zijad Mehić	Perzijsko-bosanski koledž sa internatom	100	90	0	0	0	190

18	Kemal Bećarević	JU Druga gimnazija	100	30	0	21	17	168
19	Muhamed Bećirović	Prva bošnjačka gimnazija	100	20	0	0	17	137
20	Faris Hrustić	JU Treća gimnazija	78	40	0	0	17	135
21	Inas Kasumović	JAVNA USTANOVA GIMNAZIJA OBALA Sarajevo	100	0	0	0	0	100
22	Arif Begić	JU Druga gimnazija	100	0	0	0	0	100
23	Ahmed Zaimović	Prva bošnjačka gimnazija	100	0	0	0	0	100
24	Bakir Poljčić	JU Četvrta gimnazija Ilidža	100	0	0	0	0	100
25	Ahmed Alijagić	Perzijsko-bosanski koledž sa internatom	78	0	0	8	0	86
26	FARIS SAČIĆ	JAVNA USTANOVA GIMNAZIJA OBALA Sarajevo	12	0	5	0	17	34
27	Danin Sadžak	JU Prva gimnazija Sarajevo	34	0	0	0	0	34
28	Anel Kadrić	JU SC "Nedžad Ibrišimović" Ilijaš	22	0	0	0	0	22

29	Kenan Izetbegović	JAVNA USTANOVA GIMNAZIJA OBALA Sarajevo	22	0	0	0	0	22
30	Hanan Bahtanović	Perzijsko-bosanski koledž sa internatom	22	0	0	0	0	22
31	ELDIN GUŠO	JAVNA USTANOVA GIMNAZIJA OBALA Sarajevo	0	0	5	0	17	22
32	Haris Imamović	JU Peta gimnazija	12	0	0	0	0	12
33	Din Asotić	JU Peta gimnazija	0	10	0	0	0	10
34	Adem Ajdin	JU SC "Nedžad Ibrišimović" Ilijaš	0	0	0	0	0	0
35	Adel Bajrić	JU Gimnazija Dobrinja	0	0	0	0	0	0
36	Filip Balaba	JU Gimnazija Dobrinja	0	0	0	0	0	0
37	Mak Džebo	Richmond Park College	0	0	0	0	0	0
38	Anela Delalović	Richmond Park College	0	0	0	0	0	0
39	Amin Osmanović	JU Peta gimnazija	0	0	0	0	0	0

40	Mustafa Trako	JU Četvrta gimnazija Ilidža	0	0	0	0	0	0
41	Merjem Matoruga	JU SC "Nedžad Ibrišimović" Ilijaš	0	0	0	0	0	0
42	Edis Bulbulušić	JU SC "Nedžad Ibrišimović" Ilijaš	0	0	0	0	0	0
43	HARUN RIĐEVIĆ	JAVNA USTANOVA GIMNAZIJA OBALA Sarajevo	0	0	0	0	0	0
44	Mustafa Gradišić	Perzijsko-bosanski koledž sa internatom	0	0	0	0	0	0
45	Eldar Šatara	Srednja elektrotehnička škola Sarajevo	0	0	0	0	0	0

Poredak škole

Redno mjesto	Naziv škole	Ukupno bodova
1	JU Druga gimnazija	1381
2	PU Richmond Park International Secondary School	775
3	JU Treća gimnazija	590
4	JU Četvrta gimnazija Ilidža	514
5	Prva bošnjačka gimnazija	512
6	JU Prva gimnazija Sarajevo	504
7	Perzijsko-bosanski koledž sa internatom	298
8	Srednja elektrotehnička škola Sarajevo	200
9	JAVNA USTANOVA GIMNAZIJA OBALA Sarajevo	156
10	JU Peta gimnazija	22
11	JU SC "Nedžad Ibrišimović" Ilijaš	22
12	JU Gimnazija Dobrinja	0

Postavke zadataka

1. Fabrika

Fabrika ima na raspolaganju N mašina koje moraju u narednih M dana izvršiti svaki dan po jedan zadatak. Svaka mašina ima tačno jedan zadatak za svaki dan. Znamo koliko za svaki zadatak treba minuta da ga izvrši mašina koja je za njega zadužena. Mašine mogu raditi istovremeno i jedna drugoj ne smetaju.

Naprimjer, ako imamo tri mašine, kojima treba 15, 4 i 10 minuta da izvrše svoje zadatke prvog dana, tada će nam ukupno trebati 15 minuta da se izvrše zadaci za taj dan, jer najsporija mašina mora raditi 15 minuta (druge mašine mogu svoj posao odraditi i ranije, međutim ne mogu raditi zadatak prve mašine, tako da će nam opet trebati 15 minuta). Na vama je da za svaki od M dana odredite koliko će vremena trebati da se izvrše svi dnevni zadaci.

Ulazni podaci

U prvom redu se unose prirodni brojevi N i M opisani u tekstu zadatka.
Zatim se unosi N redova po M prirodnih brojeva, ne većih od 1440, odvojenih razmakom. i-ti red predstavlja trajanje zadataka prve mašine - prvi broj u redu je trajanje zadatka (u minutama) te mašine za prvi dan, drugi broj za drugi dan i tako do kraja.

Ograničenja

$1 \leq N, M \leq 100$, svi ostali brojevi u ulazu su prirodni brojevi ne veći od 1440.

Podzadatak 1 (12 bodova)

$N = 1$

Podzadatak 2 (22 boda)

$M = 1$

Podzadatak 3 (31 bodova)

$M = N = 3$

Podzadatak 4 (35 bodova)

Bez dodatnih ograničenja.

Izlazni podaci

Program treba ispisati M brojeva. i-ti broj predstavlja broj minuta potrebnih da se završe svi zadaci i-ti dan, odnosno vrijeme potrebno da sve mašine odrade svoj zadatak. Mašine rade neovisno jedna od druge, ali svaka može raditi isključivo svoj zadatak.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
`1 5` `8 3 4 4 4`	`8 3 4 4 4`	Kako na raspolaganju imamo samo jednu mašinu tako će svaki dan posao se završiti čim ona završi svoj zadatak. To je prvi dan nakon 8 minuta, drugi dan nakon 3 minute, a treći, četvrti i peti dan nakon 4 minute. Ovaj primjer odgovara podzadatku 1.

`4 1`
`70`
`120`
`30`
`130`
	`130`	Na prvi i jedini dan četvrtoj mašini treba najviše vremena da odradi zadatak, tako da će se posao završiti nakon 130 minuta. Ovaj primjer odgovara podzadataku 2.

`3 3`
`10 6 4`
`9 7 2`
	`10 7 4`	Prvi dan dvije mašine zahtjevaju po 10 minuta, a treća 9 minuta, tako da će se svi zadaci završiti nakon 10 minuta. Drugi dan svi zadaci će se završiti za 7 minuta, pošto najsporijoj mašini (mašini 2) je potrebno 7 minuta da završi svoj zadatak. Treći dan prvoj mašini je potrebno

10 1 3		najviše vremena, a to je 4 minute. Ovaj primjer odgovara podzadatku 3.
4 4 1 2 3 4 5 6 7 8 4 4 4 4 3 9 2 8	5 9 7 8	Prvi dan trajanje zadataka mašina je 1, 5, 4 i 3 minute redom. 5 minuta je potrebno da se završe svi zadaci. Drugi dan trajanje poslova je 2, 6, 4 i 9 minuta redom. 9 minuta je potebno da se završe svi zadaci. Treći dan trajanje poslova je 3, 7, 4 i 2 minuta redom. 7 minuta je potebno da se završe svi zadaci. Četvrti dan trajanje poslova je 4, 8, 4 i 8 minuta redom. 8 minuta je potebno da se završe svi zadaci.

2. Raketne čizme

Pero će putovati stepeničastom stazom na kojoj zna ukupnu visinu svake stepenice od tla. Kako će nekada morati preći sa jedne stepenice na mnogo višu ponio je raketne čizme. Problem je što su još uvijek prototip i samo rade jednom prije nego što se pokvare.

Peri nije problem silaziti niz stepenice, ma koliko god velika razlika u visini bila. Ako iskoristi raketne čizme, Pero može preći na iduću stepenicu, ma koliko god ona viša bila od trenutne.

Vaš zadatak je da odredite koji je najveći uspon koji Pero mora napraviti sa jedne stepenice na iduću bez raketnih čizama, ako njih može iskoristiti samo jednom.

Ulazni podaci

Prva linija ulaza sadrži broj N, broj stepenica ispred Pere. Druga linija ulaza sadrži N brojeva h_i, visine stepenica ispred Pere.

Ograničenja (program se neće testirati van ovih opsega)

$1 \leq N \leq 1\,000\,000$

$-10000 \leq h_i \leq 10000$

Napomena: zbog potencijalno velikog broja upisanih brojeva predlažemo da, ukoliko koristite cin/cout za upis i ispis, na početak main funkcije dodate liniju “ios_base::sync_with_stdio(false);” kako biste ubrzali proces upisa i ispisa podataka.

Podzadatak 1 (20 bodova)

$N \leq 3$

Podzadatak 2 (20 bodova)

$N < 1000$ i postoje tačno dva para stepenica kod kojih je prva stepenica manja od naredne.

Podzadatak 3 (50 bodova)

$N < 100\,000$

Podzadatak 4 (10 bodova)

Bez dodatnih ograničenja

Izlazni podaci

Izlaz treba da se sastoji od jednog broja, najvećeg uspona koji će Pero morati napraviti bez raketnih čizama. Ukoliko Pero neće morati praviti uspon, ispisati 0.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
5 3 8 3 8 3	5	Pero će dva puta morati skakati sa niže na višu stepenicu. Raketne čizme može iskoristiti samo jednom, u ovom slučaju je svejedno za koji skok, obzirom da su oba skoka sa visinskom razlikom 5.
6 3 7 6 15 10 12	4	Ukoliko Pero u prvom skoku iskoristi raketne čizme, kasnije će morati skočiti sa stepenice visine 6 na stepenicu visine 15, što predstavlja uspon od 9. Bolje je Peri da sačuva raketne čizme za taj skok, u tom slučaju najveća visinska razlika će biti 4, što je i optimalno rješenje.

3. Biljke

U dugom, uskom stakleniku se nalazi M biljaka. Kroz tmurne i hladne zimske dane potrebno je da upalite sijalice koje će ih osvijetliti kako bi one mogle rasti. Potrebno je ovo uraditi sa minimalnom potrošnjom električne energije.

Na raspolaganju vam se nalazi N sijalica. Potrebno je da svaku biljku osvjetljava minimalno jedna sijalica. Svaka sijalica i osvjetljava sve biljke na pozicijama od A_i do B_i (uključujući i A_i i B_i), a troši C_i vati električne energije.

Potrebno je odredite minimalni ukupan trošak električne energije potreban da osvijetlite svaku biljku.

Ulazni podaci

U prvoj liniji upisa se nalaze dva broja, M i N. M je broj biljaka u stakleniku, a N je broj sijalica.

Idući red upisa se sastoji od M brojeva P_i koji predstavljaju pozicije biljaka. Idućih N redova sadrži po tri broja, A_i, B_i i C_i koji odgovaraju granicama osvjetljenog područja, te trošku energije koji odgovara sijalici broj i.

Ograničenja

$1 \leq M \leq 100, 1 \leq N \leq 20, 0 \leq A_i \leq B_i \leq 10^9, 1 \leq C_i \leq 10^9, 0 \leq P_i \leq 10^9$

Podzadatak 1 (5 bodova)

$M = 1$

Podzadatak 2 (6 bodova)

$N = 1$

Podzadatak 3 (12 bodova)

$N = 2$

Podzadatak 4 (24 bodova)

$N = 3$

Podzadatak 5 (13 bodova)

Sve sijalice na raspolaganju osvjetljavaju sve biljke.

Podzadatak 6 (19 bodova)

Sve biljke su na poziciji 0 ili 1.

Podzadatak 7 (21 bod)

Bez dodatnih ograničenja

Izlazni podaci

Ukoliko nije moguće sve biljke osvjetliti potrebno je samo ispisati -1 i time završiti rad programa.

U suprotnom, ukoliko to jeste moguće potrebno je ispisati minimalni utrošak električne energije potreban da se sve biljke osvijetle.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
3 5 0 4 6 0 7 8 0 4 3 4 4 2 4 6 4 4 6 6	7	Paljenjem druge i četvrte sijalice (utrošci 3 i 4 W, redom) osvijetlit ćemo sve tri biljke (na lokacijama 0, 4 i 6) sa utroškom energije od 7 W.

4 5 5 10 20 100 3 7 8 10 10 1 11 90 20 4 150 60 95 105 10	39	Iako bi sijalica broj 4 sama osvjetlila sve biljke (sa utroškom od 60 W) ekonomičnije rješenje je upaliti sve ostale sijalice. Na ovaj način utrošak energije je 39 W, a sve biljke su osvjetljene.
2 3 5 10 0 7 10 5 6 2 6 8 5	-1	Drugu biljku (na poziciji 10) je nemoguće osvijetliti. Ovaj primjer odgovara podzadatku 4.
3 4 1 0 0 0 0 5 2 6 1 0 3 11 1 2 6	11	Minimalni utrošak se ostvaruje paljenjem prve i posljednje sijalice, no također se ostvaruje i paljenjem samo treće sijalice. Ovaj primjer odgovara podzadatku 6.

4. Profesor Srmen

Bliži se treće i posljednje kolo SPSP-a (Sarajevskog prvenstva u sportskom penjanju). Profesor Srmen (PS) organizuje ovu trokolnu sportsku priredbu (SP) kako bi privukao svjetsku pažnju na potencijal sarajevske prirode (PSP na PSP). Super!

Poznavajući takmičare Srmen je zaključio da svaki takmičar koji je od nekog drugog imao strogo više bodova u oba dosadašnja kola sigurno neće imati strogo manje bodova od tog istog takmičara u trećem kolu. Na primjer, ako je prvi takmičar imao po 200 bodova u prva dva kola, a drugi takmičar po 100 bodova u ta ista dva kola onda drugi takmičar sigurno neće imati više bodova od prvog u trećem kolu.

Vaš zadatak je odrediti minimalno i maksimalno mjesto koje može postići svaki od takmičara pod pretpostavkom da je zaključak profesora Srmena ispravan. Mjesto se određuje na osnovu sume bodova sva tri kola. Dva takmičara sa istim brojem bodova dijele mjesto, a takmičar ostvaruje mjesto M ako i samo ako postoji tačno $M - 1$ takmičara sa više bodova od njega.

U svakom kolu je moguće ostvariti maksimalno 600 bodova, a minimalno 0.

Ulazni podaci

U prvom redu se unosi prirodan broj N, broj takmičara.

Zatim se unosi N redova koji sadrže po 2 broja od 0 do 600, a to su brojevi bodova takmičara na prvom i drugom već održanom kolu.

Ograničenja

$1 \leq N \leq 500\,000$, svi ostali brojevi u ulazu su nenegativni cijeli brojevi koji nisu veći od 600.

Napomena: zbog potencijalno velikog broja upisanih brojeva predlažemo da, ukoliko koristite cin/cout za upis i ispis, na početak main funkcije dodate liniju “ios_base::sync_with_stdio(false);” kako biste ubrzali proces upisa i ispisa podataka.

Podzadatak 1 (8 bodova)

$N = 2$

Podzadatak 2 (13 bodova)

$N = 3$

Podzadatak 3 (25 bodova)

Svi takmičari su u prva dva kola ostvarili tačno 0 ili tačno 600 bodova u bilo kojem od kola.

Podzadatak 4 (54 boda)

Bez dodatnih ograničenja

Izlazni podaci

Za svakog od N takmičara je u poseban red potrebno ispisati po 2 broja, najviše i najniže mjesto koje može postići na cjelokupnom takmičenju.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje

`2`
`250 300`
`250 250`
	`1 2`
	`1 2`
		Kako nijedan takmičar nije imao strogo više bodova od nekog drugog u oba kola tako nema ograničenja na broj bodova u trećem kolu (osim da mora biti od 0 do 600). Ovisno od broja bodova na trećem kolu oba takmičara mogu na kraju osvojiti 1. ili 2. mjesto. Ovaj primjer odgovara podzadatku 1.

`3`
`50 100`
`70 300`
`600 0`
	`2 3`
	`1 2`
	`1 3`
		Prvi takmičar sigurno ne može biti prvi jer će uvijek imati manje bodova od drugog takmičara. Slično, drugi takmičar sigurno ne može biti posljednji pošto je uvijek bolji od prvog takmičara. Treći takmičar nema nikaku garanciju za svoje konačno mjesto. Ovaj primjer odgovara podzadatku 2.

4 0 0 0 600 600 0 600 600	2 4 1 3 1 3 1 1	Ukoliko u trećem kolu prvi takmičar ima 600 bodova, a drugi i treći takmičar imaju po 0 bodova onda oni dijele 2. mjesto. Ukoliko četvrti takmičar ima 0 bodova u trećem kolu, a drugi ili treći takmičar imaju 600 onda oni mogu dijeliti 1. mjesto. Ovaj primjer odgovara podzadatku 3.
5 250 180 250 132 220 123 132 194 220 105	1 3 1 3 3 5 1 5 3 5	Prvi i drugi takmičar će biti na strogo višim pozicijama od trećeg i petog takmičara.

5. Zapetljanost

Berina voli brojeve koji su vrlo zapetljani. Ako krenemo od prirodnog broja N većeg od 1 i zamijenimo ga najmanjim prirodnim brojem kojim on nije djeljiv, te ponavljamo ovaj postupak eventualno ćemo doći do broja 2. Broj različitih brojeva koji dobijemo ovim postupkom je njegova zapetljanost. Jedini ovakav broj čija je zapetljanost 1 je upravo 2.

Berina je već napisala program koji računa zapetljanost broja, no od vas traži da riješite teži problem. Berinu zanima zbir zapetljanosti svih brojeva od A do B (uključujući i A i B).

Ulazni podaci

U prvom i jedinom redu ulaza se nalaze brojevi A i B, granice opsega za koji se treba izračunati zbir zapetljanosti.

Ograničenja

$2 \leq A \leq B \leq 10^{17}$.

Podzadatak 1 (6 bodova)

$A = B < 20$.

Podzadatak 2 (11 bodova)

$B \leq 10^{5}$.

Podzadatak 3 (18 bodova)

.$A \leq 10^{9}$, $B = A + 10^{9}$.

Podzadatak 4 (65 bodova)

Bez dodatnih ograničenja.

Izlazni podaci

Potrebno je ispisati jedan broj, zbir zapetljanosti svih brojeva u opsegu iz ulaza.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
`6 6`	`4`	Idući broj nakon 6 je 4, pa 3, pa 2. Ovo daje dužinu niza (pa samim tim i zapetljanost) jednaku 4.
`2 5`	`8`	Zapetljanosti brojeva 2, 3, 4 i 5 su 1, 2, 3 i 2, redom.
`10 100`	`235`	
`505 550`	`119`	

Rješenja zadataka

Sva rješenja, tekstovi zadataka kao i testni slučajevi su (ili će biti) dostupni na linku: https://github.com/hhadzem/ks_inf_takm.

1. Fabrika

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    int n, m;
    cin >> n >> m;

    vector <vector <int> > mat(n, vector <int>(m));
    vector <int> ispis(m, 0);

	for(int i = 0; i < n; i++)
		for(int j = 0; j < n; j++) {
			cin  >> mat[i][j];
			if(i == 0)
				ispis[i] = mat[i][j];
			ispis[j] = max(ispis[j], mat[i][j]);
		}

	for(int i = 0; i < m; i++)
		cout << ispis[i] << " ";
    return 0;
}
```

2. Raketne čizme

```
#include<iostream>
#include<vector>
#include<algorithm>
#include<limits.h>
using namespace std;

void solve_o_nlogn(vector<int> &arr, int &n) {
	vector<int> diff;
	for(int i = 1; i < n; i++)
		if(arr[i] > arr[i-1])
			diff.push_back(arr[i] - arr[i-1]);
	sort(diff.begin(), diff.end());
	cout << (diff.size() < 2 ? 0 : diff[diff.size() - 2]) << endl;
}

void solve_o_n(vector<int> &arr, int &n) {
	vector<int> diff;
	int max_first = INT_MIN, max_second = INT_MIN;
	for(int i = 1; i < n; i++)
		if(arr[i] > arr[i-1]) {
			int diff = arr[i] - arr[i-1];
			if(diff >= max_first) {
				max_second = max_first;
				max_first = diff;
			} else if(diff > max_second) {
				max_second = diff;
			}
		}
	cout << (max_second == INT_MIN ? 0 : max_second) << endl;
}

int main() {
	cin.tie(NULL);
	ios_base::sync_with_stdio(false);

	int n; cin >> n;
	vector<int> arr(n), diff;

	for(int i = 0; i < n; i++)
		cin >> arr[i];
```

```
    //solve_o_n(arr, n);
    solve_o_nlogn(arr, n);
    return 0;
}
```

3. Biljke

```
#include <iostream>
#include <vector>
#include <utility>
#include <algorithm>

using namespace std;

int main() {

    int m, n;

    cin >> m >> n;

    vector <int> vb(m), a(n), b(n), c(n);

      for(int i = 0; i < m; i++) cin >> vb[i];
      for(int i = 0; i < n; i++) cin >> a[i] >> b[i] >> c[i];

    long long min_cijena = -1, tc;
    int bitmask = 0, bmc, obm;
    vector <bool> osvjetljena(m);
    bool ok;

    while(bitmask < (1<<n)) {
        for(int i=0;i<m;i++) osvjetljena[i] = 0;
        bmc = bitmask;
        tc = 0;

        for(int i=0;i<n;i++) {
            if(bmc%2) {
                for(int j=0;j<m;j++)
                    if(vb[j] >= a[i] && vb[j] <= b[i]) osvjetljena[j] = true;
                tc += c[i];
            }
            bmc /= 2;
        }

        ok = true;
        for(int i = 0; i < m; i++)
                    ok &= osvjetljena[i];
```

```
        if(ok && (min_cijena > tc || min_cijena == -1)) {
            min_cijena = tc;
            obm = bitmask;
        }

        bitmask++;
    }

    cout << min_cijena << '\n';

    if(min_cijena == -1) return 0;

    for(int i = 0; i < n; i++) {
            cout << obm % 2;
            obm /= 2;
    }

    return 0;
}
```

4. Profesor Srmen

```cpp
#include <cstdio>
using namespace std;

const int NN = 500000;
const int MM = 600;
int a[NN], b[NN];
int k[MM + 1][MM + 1];
int s[MM + 1][MM + 1];

int suma(int p, int q, int P, int Q) {
    if (P < 0 || Q < 0)
        return 0;
    int ret = s[P][Q];
    if (p > 0)
        ret -= s[p - 1][Q];
    if (q > 0)
        ret -= s[P][q - 1];
    if (p > 0 && q > 0)
        ret += s[p - 1][q - 1];
    return ret;
}

int main() {
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; ++i) {
        scanf("%d%d", a+i, b+i);
        ++k[a[i]][b[i]];
    }
    for (int i = 0; i <= MM; ++i)
        for (int j = 0; j <= MM; ++j) {
            s[i][j] = k[i][j];
            if (i > 0)
                s[i][j] += s[i - 1][j];
            if (j > 0)
                s[i][j] += s[i][j - 1];
            if (i > 0 && j > 0)
                s[i][j] -= s[i - 1][j - 1];
        }
```

```
    for (int i = 0; i < n; ++i) {
        printf("%d ", suma(a[i] + 1, b[i] + 1, MM, MM) + 1);
        printf("%d\n", n - suma(0, 0, a[i] - 1, b[i] - 1) -
                k[0][b[i]] * (a[i] == MM) - k[a[i]][0] * (b[i] == MM));
    }
      return 0;
}
```

5. Zapetljanost

```
/*
	HONI 2012/13, 1. kolo, zadatak SNAGA
	Autor: Adrian Satja Kurdija
*/

#include <iostream>
#include <vector>
using namespace std;

int main()
{
	long long A, B, rjesenje = 0;
	cin >> A >> B;
	long long NZV = 1;

	int snaga[100];
	snaga[2] = 1;
	for (int K = 2; K < 100; K++)
	{
		for (int i = 2; i < K; i++) {
			if (K % i != 0) {
				snaga[K] = snaga[i] + 1;
				break;
			}
		}

		long long novi_NZV = NZV;
		vector<int> prosti_djelitelji_K;
		for (int p = 2, k = K; k > 1; p++) {
			if (k % p == 0) {
				prosti_djelitelji_K.push_back(p);
				while (k % p == 0) {
					k /= p;
				}
			}
		}
		if ((int)prosti_djelitelji_K.size() == 1) {
			novi_NZV *= prosti_djelitelji_K[0];
		}
```

```
            rjesenje += (snaga[K] + 1) * (B/NZV - (A-1)/NZV);

            if (novi_NZV > B || novi_NZV < 0) break;
            rjesenje -= (snaga[K] + 1) * (B/novi_NZV - (A-1)/novi_NZV);

            NZV = novi_NZV;
        }
        cout << rjesenje << endl;
}
```