



KANTON SARAJEVO
MINISTARSTVO ZA ODGOJ I OBRAZOVANJE
KANTONA SARAJEVO

Kantonalno takmičenje iz informatike za srednje škole

21. MART 2022. GODINE

Rezultati	3
Poredak takmičara	3
Poredak škole	7
Postavke zadataka	8
Termini	8
Semafor	10
Bosanski grah	13
Sedam	15
Selfie	17
Rješenja zadataka	21
Termini	21
Semafor	22
Bosanski grah	24
Sedam	25
Selfie	26



Rezultati

Poredak takmičara

Redno mjesto	Ime i prezime	Škola	Termini	Semafor	Bosanski grah	Sedam	Selfie	Total
1.	Haris Imamović	JU Druga gimnazija Sarajevo	100.0	100.0	100.0	100.0	100.0	500.0
1.	Admir Zatega	JU Druga gimnazija Sarajevo	100.0	100.0	100.0	100.0	100.0	500.0
1.	Emira Ibrahimović	JU Druga gimnazija Sarajevo	100.0	100.0	100.0	100.0	100.0	500.0
1.	Benjamin Mujkić	JU Druga gimnazija Sarajevo	100.0	100.0	100.0	100.0	100.0	500.0
1.	Faruk Ibrahimović	JU Druga gimnazija Sarajevo	100.0	100.0	100.0	100.0	100.0	500.0
6.	Muhamed Bećirović	JU Prva bošnjačka gimnazija Sarajevo	100.0	100.0	100.0	33.0	0.0	333.0
7.	Abdullah Fehratbegović	JU Druga gimnazija Sarajevo	100.0	0.0	100.0	100.0	20.0	320.0
8.	Adnan Dervišević	JU Srednja elektrotehnička škola Sarajevo	100.0	100.0	10.0	29.0	60.0	299.0
9.	Ilhan Muftić	JU Prva gimnazija Sarajevo	100.0	100.0	20.0	33.0	30.0	283.0
10.	Ervin Macić	JU Druga gimnazija Sarajevo	100.0	80.0	50.0	33.0	0.0	263.0



11.	Faris Čišija	JU Treća gimnazija Sarajevo	100.0	100.0	0.0	33.0	0.0	233.0
12.	Bakir Činjurević	JU Prva bošnjačka gimnazija Sarajevo	100.0	0.0	0.0	100.0	0.0	200.0
13.	Arif Begić	JU Druga gimnazija Sarajevo	100.0	20.0	10.0	29.0	20.0	179.0
14.	Amer Bečarević	JU Četvrta gimnazija Ilidža	100.0	0.0	50.0	22.0	0.0	172.0
15.	Harun Riđević	JU Treća gimnazija Sarajevo	100.0	0.0	30.0	33.0	0.0	163.0
16.	Zijad Mehić	Perzijsko-bosanski koledž sa internatom	100.0	20.0	0.0	22.0	0.0	142.0
17.	Tarik Bešić	JU Srednjoškolski centar "Nedžad Ibrišimović" Ilijaš	100.0	0.0	0.0	33.0	0.0	133.0
18.	Rijad Gadžo	JU Gimnazija Obala Sarajevo	100.0	0.0	0.0	29.0	0.0	129.0
19.	Muhamed Avdić	JU Četvrta gimnazija Ilidža	100.0	0.0	0.0	22.0	0.0	122.0
19.	Adnan Adrović	JU Četvrta gimnazija Ilidža	100.0	0.0	0.0	22.0	0.0	122.0
21.	Lamija Basarić	Gazi Husrev-begova medresa	95.0	0.0	0.0	22.0	0.0	117.0
22.	Eldar Šatara	JU Srednja elektrotehnička škola Sarajevo	95.0	0.0	0.0	15.0	0.0	110.0
23.	Din Asotić	JU Peta gimnazija Sarajevo	100.0	0.0	0.0	5.0	0.0	105.0
24.	Faris Sačić	JU Gimnazija Obala Sarajevo	95.0	0.0	0.0	5.0	0.0	100.0
24.	Jasenko Brakić	JU Srednja elektrotehnička škola Sarajevo	100.0	0.0	0.0	0.0	0.0	100.0



24.	Faruk Hodžić	JU Prva gimnazija Sarajevo	100.0	0.0	0.0	0.0	0.0	100.0
24.	Esmā Šabić	Gimnazija SSST	100.0	0.0	0.0	0.0	0.0	100.0
28.	Nadir Hrustanbegović	JU Treća gimnazija Sarajevo	10.0	20.0	50.0	0.0	0.0	80.0
29.	Šaćir Mešetović	JU Srednjoškolski centar "Nedžad Ibrišimović" Ilijaš	0.0	0.0	50.0	29.0	0.0	79.0
30.	Adna Mujić	JU Prva gimnazija Sarajevo	5.0	0.0	10.0	29.0	0.0	44.0
31.	Eldar Hodžić	JU Srednjoškolski centar "Nedžad Ibrišimović" Ilijaš	0.0	0.0	0.0	29.0	0.0	29.0
31.	Berin Žunić	JU Treća gimnazija Sarajevo	0.0	0.0	0.0	29.0	0.0	29.0
33.	Bakir Gracić	Gazi Husrev-begova medresa	0.0	0.0	10.0	15.0	0.0	25.0
34.	Luka Bartula	Katolički školski centar Opća-realna gimnazija Sarajevo	0.0	0.0	0.0	22.0	0.0	22.0
34.	Ahmed Alijagić	Perzijsko-bosanski koledž sa internatom	0.0	0.0	0.0	22.0	0.0	22.0
34.	Matej Slokar	Katolički školski centar Opća-realna gimnazija Sarajevo	0.0	0.0	0.0	22.0	0.0	22.0
37.	Orhan Huseinbegović	JU Peta gimnazija Sarajevo	0.0	0.0	0.0	15.0	0.0	15.0
37.	Appa Bugis Mubarak	JU Srednja elektrotehnička škola Sarajevo	0.0	0.0	0.0	15.0	0.0	15.0
39.	Naida Kaljanac	JU Prva bošnjačka gimnazija Sarajevo	5.0	0.0	0.0	0.0	0.0	5.0



39.	Mustafa Gradišić	Perzijsko-bosanski koledž sa internatom	5.0	0.0	0.0	0.0	0.0	5.0
39.	Erik Zubić	Katolički školski centar Opća-realna gimnazija Sarajevo	0.0	0.0	0.0	5.0	0.0	5.0
42.	Emir Karasalihović	JU Gimnazija Obala Sarajevo	0.0	0.0	0.0	0.0	0.0	0.0
42.	Haris Imamović	JU Peta gimnazija Sarajevo	0.0	0.0	0.0	0.0	0.0	0.0
42.	Imran Sijerčić	JU Gimnazija Dobrinja	0.0	0.0	0.0	0.0	0.0	0.0
42.	Adel Bajrić	JU Gimnazija Dobrinja	0.0	0.0	0.0	0.0	0.0	0.0
42.	Melika Mujala	Gazi Husrev-begova medresa	0.0	0.0	0.0	0.0	0.0	0.0
42.	Nadža Hasanović	Gimnazija SSST	0.0	0.0	0.0	0.0	0.0	0.0
42.	Adna Mujić	JU Prva gimnazija Sarajevo	0.0	0.0	0.0	0.0	0.0	0.0
42.	Faris Čomaga	JU Srednja mašinska tehnička škola Sarajevo	0.0	0.0	0.0	0.0	0.0	0.0
42.	Faruk Hodžić	JU Prva gimnazija Sarajevo	0.0	0.0	0.0	0.0	0.0	0.0
42.	Zara Mešević	Gimnazija SSST	0.0	0.0	0.0	0.0	0.0	0.0
42.	Amir Ljubijankić	JU Treća gimnazija Sarajevo	0.0	0.0	0.0	0.0	0.0	0.0



Poredak škole

Redno mjesto	Naziv škole	Ukupno bodova
1	JU Druga gimnazija Sarajevo	1500
2	JU Prva bošnjačka gimnazija Sarajevo	538
3	JU Srednja elektrotehnička škola Sarajevo	524
4	JU Treća gimnazija Sarajevo	476
5	JU Prva gimnazija Sarajevo	427
6	JU Četvrta gimnazija Ilidža	416
7	JU Srednjoškolski centar "Nedžad Ibrišimović" Ilijaš	241
8	JU Gimnazija Obala Sarajevo	229
9	Perzijsko-bosanski koledž sa internatom	169
10	Gazi Husrev-begova medresa	142
11	JU Peta gimnazija Sarajevo	120
12	Gimnazija SSST	100
13	Katolički školski centar Opća-realna gimnazija Sarajevo	49
14	JU Srednja mašinska tehnička škola Sarajevo	0
15	JU Gimnazija Dobrinja	0



Postavke zadatka

1. Termini

U jednoj sportskoj dvorani postoji softver za rezervisanje termina u dvorani. Međutim, šef sale zahtijeva novu stavku u softveru, a to je da za određeni dan sazna koliko je minuta slobodno u sali. Vaš je zadatak da uradite taj dio softvera.

Bit će vam na ulazu dati svi termini za taj dan, koji opisuju od kad do kad je sala zauzeta, a vi morate da na osnovu toga izračunati koliko minuta je sala slobodna.

Važno je napomenuti i to da sala radi 24h.

Ulazni podaci

U prvom redu standardnog ulaza dat je broj N - broj termina koji označavaju salu zauzetom.

U sljedećih N linija se unose po 4 broja: *start_sati*, *start_minute*, *stop_sati*, *stop_minute*.

U *start* vremenu sala postaje zauzeta a u *stop* vremenu sala više nije zauzeta.

Npr ako su *start_sati* = 13, *start_minute* = 15, *stop_sati* = 14, *stop_minute* = 5 to znači da je sala za taj termin zauzeta od 13:15 do 14:05 **ne uključujući 14:05**. To znači da je sala u 14:05 slobodna.

Ostali detalji unosa:

- Sati su dati u formatu 24h, tj, broj 13 označava 1 poslije podne.
- Nema termina koji se preklapaju.
- Svi termini su pravilnog formata (svi su u validnom vremenu i stop će uvijek biti nakon start).
- Sva vremena se odnose samo za jedan dan. Npr jedan termin ne može početi u jednom danu a završiti u drugom.
- Neće se desiti da unos za *stop* bude 00:00



- Termin ne može trajati manje od 1 minutu

Ograničenja

- $0 \leq \text{start_sati} \leq 23$,
- $0 \leq \text{start_minute}, \text{stop_minute} \leq 59$,
- $0 \leq N < 1439$

Izlazni podaci

Na standardni izlaz ispisati jednu liniju - broj minuta koliko je sala slobodna.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
5 0 0 10 0 16 0 23 0 10 0 15 0 15 0 15 45 23 15 23 45	45	Sala je slobodna u sljedećim vremenima 15min - 15:45 - 16:00 (ne uključujući trenutak 16:00 jer tad počinje novi termin) 15min - 23:00 - 23:15 (ne uključujući trenutak 23:15 jer tad počinje novi termin) 15min - 23:45 - 00:00 (ne uključujući trenutak 00:00 jer pripada sljedećem danu)
1 0 0 23 59	1	Sala je slobodna u minuti 23:59, to je jedina minuta kada je sala slobodna ovaj dan



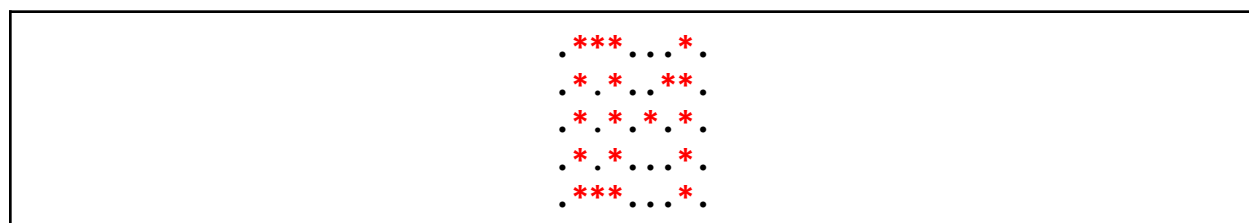
2. Semafor

Alvin je veliki ljubitelj košarke, te često odlazi na košarkaške utakmice. Kada god se nađe na utakmici, pažnju mu privlači semafor sa trenutnim rezultatom i dodatnim informacijama o igračima. Pored toga što je ljubitelj košarke, Alvin voli da se bavi statistikom i zanimaju ga informacije o brojevima koji se javljaju na semaforu. Pomozi Alvinu da prebroji koliko se puta ne semaforu pojavljuje svaka od cifara 0-9.

Svaka cifra je predstavljena u fiksnom formatu pomoću znakova “ * ” (zvjezdice) i “.” (tačkice). Sve cifre su prikazane u nastavku.

[illegible]

Semafor je predstavljen kao matrica dimenzija $\mathbf{N} \times \mathbf{M}$, pri čemu je \mathbf{N} broj redova, a \mathbf{M} broj kolona. Primjer jednog ekrana je dat ispod.



Vodite računa da je moguće da se pojedine cifre preklapaju. U tom slučaju je bez obzira na preklapanje potrebno svaku od cifara brojati pojedinačno. Također, cifra postoji na semaforu samo ukoliko je u pitanju potpuno poklapanje sa formatom datim u prvom dijelu zadatka (potrebno je da se poklapaju i zvjezdice i tačke).

Ulazni podaci

Ulaz sadrži dvije linije. U prvoj liniji se nalaze dva broja **N** i **M**, koji predstavljaju dimenzije semafora.

U narednih **N** linija se nalazi opis semafora.

Ograničenja

N i **M** su prirodni brojevi manji od 100.

Podzadatak	Ostalo	% bodova	N	M
1		20	5	3
2	Nema preklapanja cifara.	30	< 100	< 100
3	Postoji preklapanje cifara.	50	< 100	< 100

Izlazni podaci

Izlaz treba da sadrži 10 brojeva, pri čemu svaki od brojeva predstavlja koliko se puta svaka od cifara 0-9 pojavila (redom).



Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
5 9 <pre> .***. .*. .*.*.*. .*.*.*. .*.*.*. .***. .*. </pre>	1 1 0 0 0 0 0 0 0 0	Cifre 0 i 1 se pojavljuju po jednom, dok se sve ostale cifre ne pojavljuju na semaforu.
5 9 <pre> .*****. ...*.*. .***.*. ...*.*. .*****. </pre>	1 0 0 1 0 0 0 0 0 0	Cifre 0 i 3 se pojavljuju jednom, pri čemu se one poklapaju jednom od ivica.



3. Bosanski grah

Bosanski grah je staro, tradicionalno jelo koje zahtjeva posebnu pažnju prilikom pripreme. Sastojci moraju biti odabrani tako da se minimizira apsolutna razlika između kiselosti i gorčine graha. Na raspolaganju imamo N sastojaka. Svaki sastojak je opisan sa dvije vrijednosti - kiselošću i gorčinom. Kada kombinujemo nekoliko sastojaka, kiselost smjese je proizvod kiselosti sastojaka, a gorčina smjese je zbir gorčina sastojaka.

Vaš zadatak je odrediti minimalnu apsolutnu razliku između kiselosti i gorčine graha koji se može dobiti od datih sastojaka. Za grah je **potreban barem jedan od datih sastojaka**, inače grah gubi titulu bosanskoga graha.

Ulazni podaci

U prvom redu nalazi se prirodan broj N , broj sastojaka koji su nam na raspolaganju. U sljedećih N redova nalaze se po dva prirodna broja odvojena razmakom, kiselost i gorčina svakog sastojka.

Ograničenja

$$- 1 \leq N \leq 10.$$

Ako odaberemo sve sastojke, ni kiselost, ni gorčina graha neće preći 1 000 000 000.

Podzadatak	Opis podzadatka	% bodova
1	Optimalno rješenje se sastoji od samo jednog sastojka	10
2	Optimalno rješenje se sastoji od najviše dva sastojka	20
3	Optimalno rješenje se sastoji od najviše tri sastojka	20
4	Nema dodatnih ograničenja	50



Izlazni podaci

U prvi i jedini red ispišite traženu minimalnu razliku.

Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
1 3 10	7	Imamo na raspolaganju jedan sastojak, te ga moramo iskoristiti. Apsolutna razlika njegove kiselosti i gorčine je 7, pa je to i rješenje.
2 3 8 5 8	1	Ako odaberemo oba data sastojka, ukupna kiselost će biti $3 \times 5 = 15$, dok će gorčina biti $8+8=16$, i tada je tražena razlika 1. Ne može manje od ovoga.
4 1 7 2 6 3 8 4 9	1	Ako odaberemo zadnja tri sastojka, tada je ukupna kiselost jednaka $2 \times 3 \times 4 = 24$, a ukupna gorčina jednaka $6+8+9=23$. Apsolutna razlika iznosi 1, i nije moguće postići manju.



4. Sedam

Klara klaun je savladala umjetnost žongliranja i može žonglirati 7 loptica **vrlo** dugo. Pri tom žongliranju redom ciklično baca loptice - prvo u zrak baci lopticu 1, pa 2, pa 3, 4, 5, 6, 7, pa ponovo 1, pa 2, 3, 4, 5, 6, 7, 1, 2, ...

Vaš zadatak je jednostavan - odredite koju lopticu baci nakon **N** bacanja, gdje **N** može biti **vrlo** veliko.

Ulazni podaci

Ulaz sadrži samo jedan broj, **N**.

Ograničenja

N je prirodan broj. Dodatna ograničenja ovisno od podzadatka su prikazana u tabeli ispod.

Podzadatak	Ostalo	% bodova	N
1		5	≤ 7
2		10	$< 10^3$
3		7	$< 10^9$
4		7	$< 10^{18}$
5		4	$< 2^{64}$
6	Cifre su neopadajuće	40	$< 10^{1000}$
7		27	$< 10^{1000}$

Izlazni podaci

Izlaz treba da sadrži jedan broj - broj posljednje loptice koja je bačena nakon **N** bacanja.



Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
10	3	Prvih deset loptica koje se bace su 1, 2, 3, 4, 5, 6, 7, 1, 2, 3 - odakle se vidi da nakon 10. bacanja posljednja bačena loptica je loptica broj 3.
1001	7	Kako je 1001 djeljivo sa 7 tako je 1001. loptica bačena loptica broj 7.
123456789123456789123456789	1	



5. Selfie

Popularna osoba X, koja je influencer na *Kasnijegramu* (niste još čuli za ovu društvenu mrežu ali ćete čuti kasnije), uslika selfi sliku na svakom mogućem koraku od kuće do Sebilja. Neka mjesta su ljepša za slikati, neka mjesta su ružna za slikati, ali svaki korak mora biti uslikan! (*osim u autobusu*)

Osoba X je ocijenio/la cijelu mapu od svoje kuće do Sebilja, i okolo, sa ocjenom koja može biti pozitivna ili negativna, ali je cijeli broj. Veći broj predstavlja bolju ocjenu. Mapa je predstavljena kao dvodimenzionalni niz brojeva, gdje svako polje predstavlja ocjenu datu za to polje. Na toj mapi, osoba X kreće uvijek iz gornjeg lijevog ćoška - koordinata $(0, 0)$, a Sebilj se nalazi na donjem desnom ćošku - koordinata $(W-1, H-1)$. Osoba X se u svakom trenutku može kretati samo prema dole ili prema desno, osim u slučaju da će izaći iz mape, što nije dozvoljeno.

Osoba X može iskoristiti i autobus kako bi se izbjegao neki loš dio puta. Osoba X u tom slučaju će prvo uslikati polje na kojem je stajalište i ući u autobus. Osoba X neće ništa slikati u toku vožnje autobusom, ali pri izlasku će odmah uslikati izlaznu stanicu i nastaviti svoj put uz slikanje. Autobus uvijek ide u smjeru Sebilja, tj, na mapi, izlazna stanica nikad neće biti lijevo od ulazne, niti iznad.

Osoba X nije obavezna iskoristiti autobus, ukoliko je put pješke ljepši za slikanje. U većini slučajeva ovaj bus i ne postoji (pogledati *ograničenja*).

Pomozite osobi X da pronade put od kuće do Sebilja koji bi dao maksimalnu sumu svih ocjena slika sa svakog koraka tog puta. Potrebno je samo ispisati sumu ocjena na takvom putu.

Ulazni podaci

Sa standardnog ulaza se prvo unose brojevi W i H koji redom predstavljaju širinu i visinu mape. Zatim se na ulazu nalazi H redova, svaki red sadrži W brojeva G_{xy} koji predstavljaju ocjenu na polju (x,y) .



Potom se unose S_x, S_y, E_x, E_y koji predstavljaju početnu koordinatu autobuske stanice (S_x, S_y) i krajnju koordinatu autobuske stanice (E_x, E_y). **Mnogi podzadaci u zadatku nemaju busa**, tako da u slučaju da bus ne postoji na mapi za ove slučajeve su svi ovi brojevi **-1**.

Ograničenja

- $-100 \leq G_{xy} \leq 100$.

Podzadatak	Ostalo	% bodova	W	H	Autobus ?
1		10%	1	<20	NE
2		10%	1	<20	DA
3	Sva polja imaju iste ocjene	10%	<500	<500	NE
4		30%	<10	<10	NE
5		10%	<10	<10	DA
6		20%	<500	<500	NE
7		10%	<500	<500	DA

Izlazni podaci

Na jednu jedinu liniju standardnog izlaza ispisati sumu ocjena najboljeg puta.



Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
4 4 1 2 3 4 -1 -1 -1 -1 5 6 7 8 4 3 2 1 -1 -1 -1 -1	27	1 2 3 4 -1 -1 -1 -1 5 6 7 8 4 3 2 1
4 4 1 2 3 4 -1 -1 -1 -1 5 6 7 8 4 3 2 1 0 0 0 2	28	1 2 3 4 -1 -1 -1 -1 5 6 7 8 4 3 2 1 Osoba X će uzeti bus s polja (0,0) - vrijednost 1, na polje (0, 2) - vrijednost 5



3 2 1 2 3 4 5 6 -1 -1 -1 -1	16	1 2 3 4 5 6
--------------------------------------	----	----------------



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. Termini

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

// vraca koja je ovo minuta po redu u ovom danu
int minuta_po_redu(int sat, int minuta) {
    return sat * 60 + minuta;
}

int main() {
    // Krecemo s pretpostavkom da su sve minute slobodne.
    // U danu imamo 24 * 60 minuta
    int broj_slobodnih_minuta = 24 * 60;
    int n;
    cin >> n;
    for (int i=0; i<n; i++) {
        int start_sati, start_minute, stop_sati, stop_minute;
        cin >> start_sati >> start_minute >> stop_sati >> stop_minute;
        int pocetna_minuta = minuta_po_redu(start_sati, start_minute);
        int zadnja_minuta = minuta_po_redu(stop_sati, stop_minute);
        // oduzeti od ukupnog broja minuta trajanje ovog termina
        broj_slobodnih_minuta -= zadnja_minuta - pocetna_minuta;
    }
    cout << broj_slobodnih_minuta;
}
```



2. Semafor

```
#include<iostream>
#include<fstream>
#include<string>
#include<vector>

using namespace std;

int t(string a){
    int res = 0;
    for(size_t i = 0; i < a.length(); i++)
        if(a[i] == '*')
            res += (2 << i);
    return res;
}

vector<int> values(10);
void init(){
    values[0] = t("****.*.*.*.****");
    values[1] = t("..*.*.*.*..*");
    values[2] = t("***..*****.***");
    values[3] = t("***..*****.****");
    values[4] = t("*.**.*.*.*.*.*");
    values[5] = t("****.*.*.*.****");
    values[6] = t("****.*.*.*.****");
    values[7] = t("***.*..*..*..*");
    values[8] = t("****.*.*.*.****");
    values[9] = t("****.*.*.*.****");
}

int check(vector<string> &board, int row, int col){
    string s;
    for(int i = row; i < row + 5; i++)
        for(int j = col; j < col + 3; j++)
            s += board[i][j];

    int val = t(s);
    for(size_t i = 0; i < values.size(); i++)
        if(val == values[i])
            return i;
}
```



```

        return -1;
    }

    vector<int> solve(vector<string> &board){
        int N = board.size(), M = board[0].length();
        vector<int> solution(10, 0);
        for(int i = 0; i < N - 4; i++){
            for(int j = 0; j < M - 2; j++){
                int num = check(board, i, j);
                if(num == -1)
                    continue;
                solution[num] ++;
            }
        }
        return solution;
    }

    void printArray(vector<int> arr){
        for(size_t i = 0; i < arr.size(); i++)
            cout << arr[i] << " ";
        cout << endl;
    }

    int main(){
        init();

        vector<string> board;
        int N, M;
        cin >> N >> M;
        string line;
        getline(cin, line);
        while(getline(cin, line)){
            board.push_back(line);
        }
        printArray(solve(board));
        return 0;
    }

```



3. Bosanski grah

```
#include<iostream>
#include<cmath>

using namespace std;

#define MAXN 10

int n, k[MAXN], g[MAXN], best = 1000000000;

void rek( int i, int kis, int gor ) {
    if( i == n ) {
        if( gor > 0 && abs(kis-gor) < best ) best = abs(kis-gor);
    } else {
        rek( i+1, kis, gor );
        rek( i+1, kis*k[i], gor+g[i] );
    }
}

int main(void) {
    cin >> n;
    for( int i = 0; i < n; ++i )
        cin >> k[i] >> g[i];

    rek( 0, 1, 0 );

    cout<<best<<endl;
    return 0;
}
```



4. Sedam

```
#include <iostream>

using namespace std;

int main() {
    string s;
    cin >> s;

    int ostatak = 0;
    for(int i=0; i < s.length(); i++) {
        ostatak *= 10;
        ostatak += s[i] - '0';

        ostatak %= 7;
    }

    if(ostatak == 0) cout << 7;
    else cout << ostatak;

    return 0;
}
```



5. Selfie

```
#include <iostream>
#include <fstream>
using namespace std;

#define MAX_SIZE 510
int sirina, visina;
int stanica_start_x, stanica_start_y;
int stanica_kraj_x, stanica_kraj_y;

ifstream in("in.txt");
ofstream out("out.txt");

#define postavi_ret \
if (nova_vrijednost + ocjene[x][y] > ret || !ret_postavljen) { \
    ret_postavljen = true; \
    ret = nova_vrijednost + ocjene[x][y]; \
}

int rjesenjeRekurzivno(int ocjene[MAX_SIZE][MAX_SIZE], int x=sirina-1, int
y=visina-1) {
    int ret = ocjene[x][y];
    bool ret_postavljen = false;
    if (x != 0) {
        int nova_vrijednost = rjesenjeRekurzivno(ocjene, x - 1, y);
        postavi_ret;
    }
    if (y != 0) {
        int nova_vrijednost = rjesenjeRekurzivno(ocjene, x, y - 1);
        postavi_ret;
    }
    if (x == stanica_kraj_x && y == stanica_kraj_y) {
        int nova_vrijednost = rjesenjeRekurzivno(ocjene, stanica_start_x,
stanica_start_y);
        postavi_ret;
    }
    return ret;
}

int rjesenjeDP(int ocjene[MAX_SIZE][MAX_SIZE]) {
```



```

auto najbolje = new int[sirina][MAX_SIZE];
for (int x=0; x<sirina; x++) {
    for (int y=0; y<visina; y++) {
        int ret = ocjene[x][y];
        bool ret_postavljen = false;
        if (x > 0) {
            int nova_vrijednost = najbolje[x-1][y];
            postavi_ret;
        }
        if (y > 0) {
            int nova_vrijednost = najbolje[x][y-1];
            postavi_ret;
        }
        if (x == stanica_kraj_x && y == stanica_kraj_y) {
            int nova_vrijednost = najbolje[stanica_start_x][stanica_start_y];
            postavi_ret;
        }
        najbolje[x][y] = ret;
    }
}
int rezultat = najbolje[sirina-1][visina-1];
delete[] najbolje;
return rezultat;
}

int main() {
    cin >> sirina >> visina;

    int ocjene[MAX_SIZE][MAX_SIZE];

    for (int i=0; i<visina; i++) {
        for (int j=0; j<sirina; j++) {
            cin >> ocjene[j][i];
        }
    }
    cin >> stanica_start_x >> stanica_start_y;
    cin >> stanica_kraj_x >> stanica_kraj_y;
    cout << rjesenjeDP(ocjene);
}

```

