



KANTON SARAJEVO

MINISTARSTVO ZA ODGOJ I OBRAZOVANJE
KANTONA SARAJEVO

Kantonalno takmičenje iz informatike za osnovne škole

Bilten pripremili:

Hadžić Hadžem

Zavlan Amer

Džaka Tarik

Parić Muamer

26. Mart 2024. GODINE

Rezultati	3
Poredak takmičara	3
Poredak škola	5
Postavke zadataka	6
Zadatak 1 - Hobbit	6
Zadatak 2 - Jedrilica	8
Zadatak 3 - Dronovi	10
Zadatak 4 - Alinina abeceda	12
Zadatak 5 - Kotrljanje	14
Rješenja zadataka	16
Zadatak 1 - Hobbit	16
Zadatak 2 - Jedrilica	17
Zadatak 3 - Dronovi	18
Zadatak 4 - Alinina abeceda	21
Zadatak 5 - Kotrljanje	24



Rezultati

Poredak takmičara

Mjes to	Ime i prezime	Škola	Z1	Z2	Z3	Z4	Z5	Ukup no
1	Amir Karamović	JU OŠ "Musa Ćazim Ćatić"	100	100	100	100	0	400
2	Tarik Odžak	OŠ "Grbavica1"	100	100	100	0	0	300
3	Faris Šabeta	JU OŠ "Musa Ćazim Ćatić"	100	100	0	20	0	220
4	Tarik Ajanović	OŠ "Zahid Baručija"	100	100	0	0	0	200
5	Mak Nurboja	JU OŠ "Malta"	100	100	0	0	0	200
6	Biogradlić Afan	OŠ "Saburina"	100	100	0	0	0	200
7	Emin Džambegović	JU "Prva osnovna škola"	100	100	0	0	0	200
8	Isa Kalajac	OŠ "Hrasno"	100	100	0	0	0	200
9	Omer Drkić	JU "Druga osnovna škola" Ilidža	100	15	25	25	0	165
10	Hafizagić Ahmed	OŠ "Saburina"	100	15	25	20	0	160
11	Jan Lučić	OŠ "Grbavica1"	100	5	0	25	0	130
12	Malik Roman Hadžibulić	OŠ "Silivije Strahimir Kranjčević"	100	0	0	25	0	125
13	Borna Mraović	OŠ "Osman Nuri Hadžić"	100	5	0	0	0	105
14	Ema Drljević	JU OŠ "Musa Ćazim Ćatić"	100	5	0	0	0	105
15	Almin Šarić	OŠ "6. mart" Hadžići	100	0	0	0	0	100
16	Harun Tukulija	JU OŠ "Malta"	100	0	0	0	0	100
17	Dilan Tuzović	OŠ "Hrasno"	100	0	0	0	0	100
18	Lamija Avdić	OŠ "Zahid Baručija"	100	0	0	0	0	100
19	Arijan Stanojlović	OŠ "Meša Selimović"	100	0	0	0	0	100
20	Mahir Dupovac	OŠ "6. mart" Hadžići	100	0	0	0	0	100



21	Ahmed Kamerić	JU OŠ "Malta"	100	0	0	0	0	100
22	Samra Kulenović	OŠ "6. mart" Hadžići	100	0	0	0	0	100
23	Ahmed Topčić	OŠ "Meša Selimović"	100	0	0	0	0	100
24	Hamza Alihodžić	OŠ "Đulistan"	100	0	0	0	0	100
25	Nadir Čaluk	OŠ "Osman Nuri Hadžić"	100	0	0	0	0	100
26	Tarik Voloder	JU "Druga osnovna škola" Ilidža	100	0	0	0	0	100
27	Harun Zahirović	OŠ "Đulistan"	100	0	0	0	0	100
28	Ajša Kapetanović	OŠ "Meša Selimović"	100	0	0	0	0	100
29	Vedad Zulum	OŠ "Hrasno"	100	0	0	0	0	100
30	Ajna Hadžić	JU "Druga osnovna škola" Ilidža	100	0	0	0	0	100
31	Njemčević Tahir	OŠ "Saburina"	0	0	0	0	0	0
32	Teo Kočo	OŠ "Silivije Strahimir Kranjčević"	0	0	0	0	0	0
33	Sina Letuka	OŠ "Zahid Baručija"	0	0	0	0	0	0
34	Fatih Prašović	JU "Prva osnovna škola"	0	0	0	0	0	0
35	Džan Prašović	JU "Prva osnovna škola"	0	0	0	0	0	0
36	Bilal Alihodžić	OŠ "Đulistan"	0	0	0	0	0	0
37	Harun Demirović	OŠ "Osman Nuri Hadžić"	0	0	0	0	0	0
38	Ahmed Kozarić	OŠ "Grbavica1"	0	0	0	0	0	0
39	Ajlin Arifović	OŠ "Silivije Strahimir Kranjčević"	0	0	0	0	0	0



Poredak škola

Redno mjesto	Naziv škole	Ukupno bodova
1	JU OŠ "Musa Ćazim Ćatić"	725
2	OŠ "Grbavica1"	430
3	JU OŠ "Malta"	400
4	OŠ "Hrasno"	400
5	JU "Druga osnovna škola" Ilidža	365
6	OŠ "Saburina"	360
7	OŠ "Zahid Baručija"	300
8	OŠ "Meša Selimović"	300
9	OŠ "6. mart" Hadžići	300
10	OŠ "Osman Nuri Hadžić"	205
11	OŠ "Đulistan"	200
12	JU "Prva osnovna škola"	200
13	OŠ "Silivije Strahimir Kranjčević"	125



Postavke zadatka

Zadatak 1 - Hobbit

Nakon uspješne sezone zajedničkog uzgajanja u Shire-u, Merry i Pippin su uzgojili N kg krompira. Odlučili su da sav krompir koji su uzgojili dadnu prijatelju Samwise-u. Problem im predstavlja što posjeduju samo vreće u koje može stati tačno M kg krompira, pri čemu žele da je svaka vreća puna do vrha kada je budu davali Samwise-u (u vreći je tačno M kg krompira). Vaš zadatak je da provjerite da li su u mogućnosti to uraditi.

Kao pomoć u rješavanju ovog zadatka, prisjećate se načina ispitivanja djeljivosti broja u C++:

```
if(A % B == 0) cout << "Broj A je djeljiv sa B";  
else cout << "Broj A nije djeljiv sa B";
```

i Python programskom jeziku:

```
if A % B == 0:  
    print("A je djeljivo sa B")  
else:  
    print("A nije djeljivo sa B")
```

Ulazni podaci

U prvom redu ulaza se nalazi N - broj kilograma krompira koji su Merry i Pippin uzgojili.

U drugom redu ulaza se nalazi M - broj kilograma krompira koji može stati u vreću.

Ograničenja

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



Izlazni podaci

U prvi i jedini red izlaza potrebno je ispisati “MOGU” ako Merry i Pippin mogu na željeni način Samwise-u dati tačno trećinu krompira, odnosno potrebno je ispisati “NE MOGU”, u suprotnom.

Primjer

Ulaz	Očekivani izlaz	Objašnjenje
15 5	MOGU	Merry i Pippin mogu napuniti 3 vreće sa tačno 5 kg krompira i dati ih Samwise-u, čime mu daju sav krompir koji su uzgojili.
20 3	NE MOGU	Nema načina na koji mogu dati sav krompir koristeći vreće od tačno 3 kg. U nekoj vreći će morati biti različita količina krompira.
108 3	MOGU	Željenih 36 kg krompira mogu rasporediti u 36 vreća po 3 kg, te takve vreće dati prijatelju, čime su ispunili cilj.



Zadatak 2 - Jedrilica

Pero uživa u zasluženom odmoru na svojoj jedrilici. Iako je na odmoru, iz dosade je odlučio analizirati svoje jučerašnje kretanje jedrilicom. Njegov kompas povremeno spašava smjer kretanja njegovog broda.

Smjer kretanja je cijeli broj između 0 i 359. 0 označava *sjever*, 90 označava *istok*, 180 označava *jug*, a 270 označava *zapad*. Svaki podatak koji kompas spasi na ovaj način nazivamo **očitanjem smjera**. Pero na raspolaganju ima N očitavanja smjera koje je njegov kompas jučer zabilježio.

Kada posmatramo dva uzastopna očitavanja smjera S_1 i S_2 , iz S_1 je moguće doći u S_2 na dva načina: pomjeranjem broda u smjeru kazaljke na satu ili u obrnutom smjeru od kretanja kazaljke na satu. Pero se ne može uvijek sjetiti koji smjer je odabrao, pa će pretpostaviti da je uvijek birao smjer koji mu omogućava **manje skretanje**. Npr. ukoliko su dva uzastopna očitavanja smjera jednaka 100 i 200, Pero pretpostavlja da je rotirao broj u smjeru kazaljke na satu praveći tako skretanje od 100 stepeni. Međutim, ako su dva uzastopna očitavanja smjera jednaka 100 i 300, Pero pretpostavlja da se kretao u smjeru suprotnom od kazaljke na satu i tako je napravio očitavanje smjera od 160 stepeni. Također, Pero pretpostavlja da između dva uzastopna očitavanja smjera nikad nije mijenjao svoj smjer rotacije broda.

Ukupno skretanje broda predstavlja zbir svih skretanja koje je Pero napravio. Pomozite mu da izračuna ovu vrijednost tako što ćete napisati program koji za data očitavanja smjera ispisuje ukupno skretanje puta.

Ulazni podaci

U prvoj liniji ulaza unosi se cijeli broj N koji predstavlja broj očitavanja smjera koje je kompas zabilježio. U svakoj od sljedećih N linija, nalazi se po jedan cijeli broj između 0 i 359, koji predstavljaju očitavanja smjera.

Ograničenja

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



Izlazni podaci

Izlaz treba da sadrži **samo** jedan broj - ukupno skretanje jučerašnjeg puta.

Primjer

Ulaz	Očekivani izlaz	Objašnjenje
5 0 5 10 15 20	20	Pero je počeo u smjeru 0, a završio u smjeru 20, pri tome se pomjerajući za 5 stepeni u smjeru kazaljke na satu. Ukupno skretanje puta je očigledno 20.
3 300 200 100	200	Pero se kreće u smjeru suprotnom od smjera kazaljke na satu i to dva puta po 100 stepeni, praveći tako ukupno skretanje puta od 200 stepeni.
5 5 20 330 270 290	145	Pero prvi napravi skretanje od 15 stepeni prilikom rotiranja broda sa 5 na 20 stepeni. Zatim napravi skretanje u smjeru suprotnom od kazaljke na satu sa 20 stepeni na 330, što je skretanje od 50 stepeni. Onda nastavlja u istom smjeru do 270 stepeni, što je ponovo 60 stepeni. Konačno, ponovo mijenja smjer i skretanjem od 20 stepeni završi finalno u smjeru od 290 stepeni. Ukupno skretanje puta: $15 + 50 + 60 + 20 = 145$.



Zadatak 3 - Dronovi

Mujo i Huso su napravili svoje dronove od Lego kockica i igraju se sa njima. Kako bi ih kontrolisali zadaju im niz naredbi sa smjerom kretanja i udaljenosti koju će preći u tom smjeru, u metrima.

Prvo postave dronove tako da je Husin dron jedan metar desno od Mujinog. Dronovi polete i izvrše se komande koje su im zadali, te slete. Vaš zadatak je nakon tih komandi odrediti koliko su udaljeni Mujin i Husin dron. Mujo i Huso su omogućili 8 smjerova kretanja sa svojim oznakama, i to:

- S - sjever
- SI - sjeveroistok
- I - istok
- JI - jugoistok
- J - jug
- JZ - jugozapad
- Z - zapad
- SZ - sjeverozapad

Komande su zadane u formatu “[smjer] [udaljenost]” gdje je smjer jedna od skraćenica iz liste iznad, a udaljenost pozitivan broj manji od 1000 koji nije nužno cijeli. Nakon upisa potrebno je ispisati udaljenost između dronova.

Ulazni podaci

U prvom redu ulaza se nalazi M , broj Mujinih naredbi.

U idućih M redova se nalaze naredbe za Mujin dron. Prvo se nalazi skraćenica za smjer, pa razmak, pa decimalni broj D_i koji označava vrijednost koju je dron prešao u tom smjeru, u metrima.

U idućem redu se nalazi broj H , broj Husinih naredbi.

U idućih H redova se nalaze naredbe za Husin dron u istom obliku kao i za Mujin dron.

Ograničenja

$$1 \leq M, H, D_i \leq 1\,000$$



Izlazni podaci

Potrebno je ispisati **samo** jedan decimalni broj, udaljenost između konačnih pozicija dronova, u metrima. Greška ne smije biti veća od jednog milimetra (0.001 metara).

Primjer

Ulaz	Očekivani izlaz	Objašnjenje
2 S 1 I 1 4 J 2 I 2 S 2 Z 2	1	Mujin dron je počeo 1 metar zapadno od početne pozicije Husinog, a završio 1 metar sjeverno od nje. Husin drno se vratio na svoju početnu tačku.
3 JZ 3 I 2 JZ 1.5 2 JI 1.25 SZ 2.45	4.24536	Ukoliko pretpostavimo da je početna pozicija Husinog drona bila $x = 1, y = 0$, a Mujinog drona $x = 0, y = 0$, tada je krajnja pozicija Husinog drona približno $x = 0.151472, y = 0.848528$, dok je krajnja lokacija Mujinog drona približno: $x = -1.18198, y = -3.18198$. Udaljenost između ove dvije tačke je: 4.24536.



Zadatak 4 - Alinina abeceda

Alina je napravila svoj jezik sa svojom abecedom. Njena abeceda se sastoji od istih slova kao i engleska abeceda, ali riječi su poredane drugačijim redoslijedom. Alina će vam reći redoslijed slova u njenoj abecedi i dati vam niz od N riječi, a vaš zadatak je da poredate te riječi onako kako bi bile i u rječniku. Ovo znači da:

- Ako dvije riječi počinju sa različitim slovom tada se riječ koja počinje sa ranijim slovom abecede pojavljuje ranije u rječniku.
- Ako dvije riječi imaju prvih nekoliko slova zajedničkih onda se međusobni poredak određuje na osnovu prvog različitog slova. Riječ sa ranijim slovom iz abecede se pojavljuje prije.
- Ako je jedna riječ potpuno sadržana u početku druge tada se kraća riječ pojavljuje prije (riječ “kap” se pojavljuje prije riječi “kapa”).

Prvo slovo svake riječi će biti veliko; redoslijed velikih slova je isti kao redoslijed malih slova.

Ulazni podaci

U prvom redu ulaza se nalazi svih 26 slova engleske abecede, bez razmaka između njih.

U drugom redu se nalazi N - broj riječi koje trebate međusobno poredati.

U svakom od idućih N redova se nalazi riječ dužine D_i iz niza riječi.

Ograničenja

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

Izlazni podaci

Potrebno je ispisati N redova koji sadrže upisane riječi, redoslijedom na osnovu Alinine abecede.



Primjer

Ulaz	Očekivani izlaz	Objašnjenje
abcdefghijklmnopq rstuvwxyz 5 Tastatura Mis Monitor Punjac Laptop	Laptop Mis Monitor Punjac Tastatura	Alinina abeceda je jednaka engleskoj abecedi, samim time poredak je isti kao u standardnim rječnicima.
zyxwvutsrqponmlkj ihgfedcba 5 Audi Mercedes Opel Toyota Volkswagen	Volkswagen Toyota Opel Mercedes Audi	Alinina abeceda je praktično samo obrnuta standardna abeceda, pa je potrebno poredati riječi u suprotnom redoslijedu od prirodnog.
wakucyztfnqijxrohgd mlpvbse 10 Moj Sarafciger Je Najsarafcigerastiji Sarafciger Medju Najsarafcigerastiji m Sarafcigerima Na Svijetu	Na Najsarafciger astiji Najsarafciger astijim Je Moj Medju Sarafciger Sarafciger Sarafcigerima Svijetu	Sve riječi počinju sa jednim od četiri slova "J", "N", "M", "S". Potrebno je iz Alinine abecede analizirati redoslijed ovih slova. Kao što vidimo redoslijed je "N"- "J"- "M"- "S". Nakon toga za riječi koje počinju slovom "M", moramo analizirati redoslijed slova "E" i "O", kao što vidimo redoslijed je "O"- "E". Konačno, za riječi koje počinju slovom "S", analiziramo redoslijed slova "A" i "V", i kao što vidimo redoslijed je "A"- "V". Zatim primijenjujemo i preostalo pravilo o dužini riječi.



Zadatak 5 - Kotrljanje

Na ploču sa brojevima postavljena je kocka. Polja brojeva su velika kao i kocka, tako da jedna strana kocke savršeno pokriva jedan broj. Kocka se kotrlja po ploči i svakim korakom se kocka pomjeri iznad novog (susjednog) broja, te se strana kocke koja dodiruje ploču također promijeni. Svaka strana kocke pamti zbir svih brojeva koje je dotakla. Kocka nikada neće otići van opsega ploče.

Vaš zadatak je da ispišete zbirove sa svih 6 stranica kocke, poredane od najvećeg zbira do najmanjeg.

Ulazni podaci

Prvi red ulaza se sastoji od 2 broja, M i N , broj redova i broj kolona ploče.

Idućih M redova se sastoji od po N brojeva, B_{ij} odvojenih razmakom, koji predstavljaju brojeve na ploči.

Naredni red sadrži 2 broja, R_p i K_p , početni red i kolonu kocke.

Naredni red sadrži 1 broj, D broj koraka koje je kocka napravila.

Konačno, posljednji red sadrži D slova "S", "I", "Z" ili "J", koji predstavljaju smjer u kojem se kocka kotrljala ("S" - sjever, "I" - istok, "Z" - zapad, "J" - jug).

Ograničenja

$$1 \leq M, N \leq 500$$

$$1 \leq B_{ij} \leq 1000$$

$$1 \leq R_p \leq M$$

$$1 \leq K_p \leq N$$

$$1 \leq D \leq 10\,000$$

Izlazni podaci

Potrebno je ispisati vrijednosti na stranama kocke, poredane u opadajućem redoslijedu, svaka u odvojenom redu.



Primjer

Ulaz	Očekivani izlaz	Objašnjenje
<pre> 3 3 1 2 3 4 5 6 7 8 9 2 2 2 I J </pre>	<pre> 9 6 5 0 0 0 </pre>	<p>U početnom trenutku kocka se nalazi iznad polja sa vrijednošću 5. Zatim se kreće istočno i dolazi iznad polja sa vrijednošću 6. Konačno se kreće južno i dolazi na polje sa vrijednošću 9. Ostale stranice kocke nisu dodirivale ploču, pa su njihove vrijednosti 0.</p>
<pre> 2 6 1 2 3 4 5 6 7 8 9 10 11 12 1 1 7 J I I I I I S </pre>	<pre> 20 18 10 9 7 0 </pre>	<p>Ako stranice kocke numerišemo brojevima 1 - 6, tada je svaka od stranica dodirivala sljedeća polja:</p> <p>1 -> 1, 6 = 7 2 -> 7, 11 = 18 3 -> 8, 12 = 20 4 -> 9 = 9 5 -> 10 = 10 6 -> 0 = 0</p>



Rješenja zadataka

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

Zadatak 1 - Hobbit

C++

```
#include<iostream>
using namespace std;
int main() {
    int N, M;
    cin >> N >> M;
    if(N % M == 0) cout << "MOGU"; else cout << "NE MOGU";
    return 0;
}
```

Python

```
N = int(input())
M = int(input())
if N % M == 0:
    print("MOGU")
else:
    print("NE MOGU")
```



Zadatak 2 - Jedrilica

C++

```
#include<iostream>
#include<vector>
using namespace std;
int main() {
    int N; cin >> N;
    vector<int> arr(N);
    for(int i = 0 ; i < N; i++)
        cin >> arr[i];
    int result = 0;
    for(int i = 0; i < N - 1; i++) {
        int diff = max(arr[i], arr[i+1]) - min(arr[i], arr[i+1]);
        if(diff > 180) diff = 360 - diff;
        result += diff;
    }
    cout << result << endl;
    return 0;
}
```

Python

```
n = int(input())
lista = []
for i in range(n):
    lista.append(int(input()))

result = 0
for i in range(n - 1):
    diff = max(lista[i], lista[i+1]) - min(lista[i], lista[i+1])
    if diff > 180: diff = 360 - diff
    result += diff

print(result)
```



Zadatak 3 - Dronovi

C++

```
#include<iostream>
#include<math.h>
using namespace std;

void parse(string dir, float &x_sign, float &y_sign) {
    if(dir == "S") { x_sign = 0; y_sign = 1; }
    else if(dir == "J") { x_sign = 0; y_sign = -1; }
    else if(dir == "I") { x_sign = 1; y_sign = 0; }
    else if(dir == "Z") { x_sign = -1; y_sign = 0; }
    else if(dir == "SI") { x_sign = 1./sqrt(2); y_sign = 1./sqrt(2); }
    else if(dir == "SZ") { x_sign = -1./sqrt(2); y_sign = 1./sqrt(2); }
    else if(dir == "JI") { x_sign = 1./sqrt(2); y_sign = -1./sqrt(2); }
    else if(dir == "JZ") { x_sign = -1./sqrt(2); y_sign = -1./sqrt(2); }
}

int main() {
    int M, H;
    float xm = 0, ym = 0;
    float xh = 1, yh = 0;

    cin >> M;
    for(int i = 0 ; i < M; i++) {
        string dir; float value;
        cin >> dir >> value;
        float x_sign, y_sign;
        parse(dir, x_sign, y_sign);
        xm += x_sign * value;
        ym += y_sign * value;
    }

    cin >> H;
    for(int i = 0 ; i < H; i++) {
        string dir; float value;
```



```

    cin >> dir >> value;
    float x_sign, y_sign;
    parse(dir, x_sign, y_sign);
    xh += x_sign * value;
    yh += y_sign * value;
}

float diff = sqrt((xm - xh) * (xm - xh) + (ym - yh) * (ym - yh));
cout << diff << endl;
return 0;
}

```

Python

```

import math
def parse(dir):
    if dir == "S": return 0, 1
    elif dir == "J": return 0, -1
    elif dir == "I": return 1, 0
    elif dir == "Z": return -1, 0
    elif dir == "SI": return 1/math.sqrt(2), 1/math.sqrt(2)
    elif dir == "SZ": return -1/math.sqrt(2), 1/math.sqrt(2)
    elif dir == "JI": return 1/math.sqrt(2), -1/math.sqrt(2)
    elif dir == "JZ": return -1/math.sqrt(2), -1/math.sqrt(2)

xm, ym = 0.0, 0.0
xh, yh = 1.0, 0.0

M = int(input())
for i in range(M):
    dir, value = input().split()
    value = float(value)

    x_sign, y_sign = parse(dir)
    xm += x_sign * value
    ym += y_sign * value

H = int(input())

```



```
for i in range(H):
    dir, value = input().split()
    value = float(value)

    x_sign, y_sign = parse(dir)
    xh += x_sign * value
    yh += y_sign * value

diff = math.sqrt((xm - xh) ** 2 + (ym - yh) ** 2)
print(diff)
```



Zadatak 4 - Alinina abeceda

C++

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

string alphabet;
vector<int> val(26);

bool compare(const string &a, const string &b) {
    if(a.length() > b.length()) return !compare(b, a);

    for(int i = 0; i < a.length(); i++) {
        if(a[i] == b[i]) continue;

        char AA = 'a';
        if(i == 0) AA = 'A';

        return val[a[i]-AA] < val[b[i]-AA];
    }

    if(a.length() == b.length()) return false;

    return true;
}

int main() {
    cin >> alphabet;
    for(int i = 0 ; i < alphabet.length(); i++) {
        char curr = alphabet[i];
        val[curr - 'a'] = i;
    }

    int n; cin >> n;
    vector<string> words(n);
    for(int i = 0; i < n; i++)
        cin >> words[i];
}
```



```

sort(words.begin(), words.end(), compare);

for(int i = 0 ; i < n; i++)
    cout << words[i] << endl;
return 0;
}

```

Python

```

def compare(a, b):
    if len(a) > len(b):
        return not compare(b, a)

    for i in range(len(a)):
        if a[i] == b[i]:
            continue

        AA = 'a'
        if i == 0:
            AA = 'A'
        return val[ord(a[i]) - ord(AA)] < val[ord(b[i]) - ord(AA)]

    if len(a) == len(b):
        return False
    return True

if __name__ == "__main__":
    alphabet = input().strip()
    val = [0] * 26
    for i, curr in enumerate(alphabet):
        val[ord(curr) - ord('a')] = i

    n = int(input().strip())
    words = []
    for _ in range(n):
        words.append(input().strip())

```



```
for i in range(n):
    for j in range(i + 1, n):
        if not compare(words[i], words[j]):
            words[i], words[j] = words[j], words[i]

for word in words:
    print(word)
```



Zadatak 5 - Kotrljanje

C++

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

void t_zapad(vector<int> &conf) {
    int temp = conf[0];
    conf[0] = conf[1]; conf[1] = conf[2]; conf[2] = conf[3]; conf[3] = temp;
}

void t_istok(vector<int> &conf) {
    int temp = conf[0];
    conf[0] = conf[3]; conf[3] = conf[2]; conf[2] = conf[1]; conf[1] = temp;
}

void t_jug(vector<int> &conf) {
    int temp = conf[1];
    conf[1] = conf[5]; conf[5] = conf[3]; conf[3] = conf[4]; conf[4] = temp;
}

void t_sjever(vector<int> &conf) {
    int temp = conf[1];
    conf[1] = conf[4]; conf[4] = conf[3]; conf[3] = conf[5]; conf[5] = temp;
}

int main() {
    int M, N;
    cin >> M >> N;

    vector<vector<int>> B(M, vector<int>(N));
    for(int i = 0; i < M; i++)
        for(int j = 0; j < N; j++)
            cin >> B[i][j];

    int start_x, start_y;
    cin >> start_x >> start_y;
```




```

start_x--; start_y--;

int k;
cin >> k;

vector<char> arr(k);
for(int i = 0 ; i < k; i++)
    cin >> arr[i];

vector<int> values(6, 0);
vector<int> conf{0, 1, 2, 3, 4, 5};
values[1] += B[start_x][start_y];

for(int i = 0 ; i < k; i++) {
    if(arr[i] == 'J') { t_jug(conf); start_x += 1; }
    else if(arr[i] == 'S') { t_sjever(conf); start_x -= 1; }
    else if(arr[i] == 'I') { t_istok(conf); start_y += 1; }
    else { t_zapad(conf); start_y -= 1; }
    values[conf[1]] += B[start_x][start_y];
}

sort(values.rbegin(), values.rend());
for(auto el : values)
    cout << el << " ";
return 0;
}

```

Python

```

def t_zapad(conf):
    conf[0], conf[1], conf[2], conf[3] = conf[1], conf[2], conf[3], conf[0]

def t_istok(conf):
    conf[0], conf[3], conf[2], conf[1] = conf[3], conf[2], conf[1], conf[0]

def t_jug(conf):
    conf[1], conf[5], conf[3], conf[4] = conf[5], conf[3], conf[4], conf[1]

def t_sjever(conf):
    conf[1], conf[4], conf[3], conf[5] = conf[4], conf[3], conf[5], conf[1]

if __name__ == "__main__":

```



```

M, N = map(int, input().split())

B = []
for _ in range(M):
    row = list(map(int, input().split()))
    B.append(row)

start_x, start_y = map(int, input().split())
start_x -= 1
start_y -= 1

k = int(input())
arr = input().split()

values = [0] * 6
conf = [0, 1, 2, 3, 4, 5]
values[1] += B[start_x][start_y]

for i in range(k):
    if arr[i] == 'J':
        t_jug(conf)
        start_x += 1
    elif arr[i] == 'S':
        t_sjever(conf)
        start_x -= 1
    elif arr[i] == 'I':
        t_istok(conf)
        start_y += 1
    else:
        t_zapad(conf)
        start_y -= 1
    values[conf[1]] += B[start_x][start_y]

values.sort(reverse=True)
print(*values)

```

