



KANTON SARAJEVO  
MINISTARSTVO ZA ODGOJ I OBRAZOVANJE  
KANTONA SARAJEVO

---

# Kantonalno takmičenje iz informatike za osnovne škole

---

26. MART 2023. GODINE

<b>Rezultati</b>	<b>3</b>
Poredak takmičara	3
Poredak škole	8
<b>Postavke zadataka</b>	<b>9</b>
1. Magični kvadrat	9
2. Drvosječa	12
3. Broj palindrom	15
4. Mrežopolis	17
5. Isamovski brojevi	20
<b>Rješenja zadataka</b>	<b>22</b>
1. Magični kvadrat	22
2. Semafor	24
3. Broj palindrom	25
4. Mrežopolis	27
5. Isamovski brojevi	28



# Rezultati

## Poredak takmičara

Redno mjesto	Ime i prezime	Škola	Magični kvadrat	Drvosječa	Broj palindrom	Mrežopolis	Isamovski brojevi	Total
1	Adnan Osmić	JU OŠ "Musa Ćazim Ćatić"	100	100	100	100	34	434
2	Alen Avdibegović	JU OŠ "Musa Ćazim Ćatić"	100	100	51	100	34	385
3	Amir Karamović	JU OŠ "Musa Ćazim Ćatić"	100	100	6	33	9	248
4	Tarik Dacić	JU OŠ "Pofalići"	100	100	26	0	21	247
5	Mirza Bušatlić	JU Osnovna škola "Grbavica II"	100	85	51	0	9	245
6	Mak Čelan	JU OŠ "ALEKSA ŠANTIĆ"	100	100	6	0	9	215
7	Nermin Malićević	JU "DESETA OSNOVNA ŠKOLA" ILIDŽA	100	45	26	0	21	192
8	Hamza Banjanović	JU Osman Nuri Hadžić	100	45	26	0	21	192
9	Zinedin Hadžić	JU OŠ „Čengić Vila 1“	100	30	26	0	9	165



10	Merima Gaštan	JU OŠ „Čengić Vila 1“	100	30	26	0	9	165
11	Tarik Melunović	JU OŠ „Čengić Vila 1“	100	45	0	0	9	154
12	Emin Džambegović	Prva osnovna škola Ilidža	100	45	0	0	9	154
13	Alen Kerić	Richmond Park Primary School Sarajevo	100	30	6	0	9	145
14	Taha Žgalj	JU OŠ "Hamdija Kreševljaković" Sarajevo	100	30	6	0	9	145
15	Tarik Amar Sućeska	JU Osman Nuri Hadžić	100	30	6	0	9	145
16	Vedad Termiz	Richmond Park Primary School Sarajevo	100	30	0	0	0	130
17	Imad Džigal	JU OŠ "ALEKSA ŠANTIĆ"	100	30	0	0	0	130
18	Neila Rizvo	JU OŠ "9. maj" Pazarić, Sarajevo	100	30	0	0	0	130
19	Amna Zukanović	Prva osnovna škola Ilidža	100	30	0	0	0	130
20	Faris Makaš	JU OŠ "Skender Kulenović"	100	30	0	0	0	130
21	Asja Čišija	JU OŠ "Skender Kulenović"	100	30	0	0	0	130
22	Demirović Nedžla	JU Osman Nuri Hadžić	100	30	0	0	0	130



23	Bakir Kabulović	JU OŠ "9. maj" Pazarić, Sarajevo	100	10	6	0	9	125
24	Nidal Hastor	JU Osnovna škola "Grbavica II"	100	10	6	0	9	125
25	Benjamin Katić	JU OŠ "Skender Kulenović"	100	20	0	0	0	120
26	Edi Lukić	Ćamil Sijarić	100	0	6	0	9	115
27	Omar Hasanović	Osnovna škola "Izet Šabić"	100	0	0	0	9	109
28	Hamza Oprašić	Osnovna škola "Izet Šabić"	49	10	26	0	21	106
29	Hanan Šarić	JU OŠ "9. maj" Pazarić, Sarajevo	100	0	0	0	0	100
30	Aldin Subašić	Ćamil Sijarić	100	0	0	0	0	100
31	Faris Zilić	JU OŠ "Pofalići"	100	0	0	0	0	100
32	Ajna Kulenović	Ćamil Sijarić	100	0	0	0	0	100
33	Hena Ahmetspahić	JU OŠ "ALEKSA ŠANTIĆ"	100	0	0	0	0	100
34	Emina Moćević	JU OŠ "Pofalići"	100	0	0	0	0	100
35	Ishak Mahmutović	Prva osnovna škola Ilidža	18	30	26	0	9	83



36	Abdullah Muslić	JU"Peta osnovna škola"Sokolovići	49	10	0	0	0	59
37	Faris Spahić	Osnovna škola "Izet Šabić"	49	0	0	0	0	49
38	Asija Avdić	JU "DESETA OSNOVNA ŠKOLA" ILIDŽA	18	30	0	0	0	48
39	Rijana Hukara	JU "DESETA OSNOVNA ŠKOLA" ILIDŽA	0	30	0	0	0	30
40	Ismail Zeherović	OŠ "Đulistan" Lješevo, Ilijaš	18	0	0	0	0	18
41	Ammar Kabaši	JU Osnovna škola "Grbavica II"	0	10	0	0	0	10
42	Hana Tahmiščija	JU OŠ "Silvije Strahimir Kranjčević"	0	10	0	0	0	10
43	Malik Roman Hadžibulić	JU OŠ "Silvije Strahimir Kranjčević"	0	0	0	0	0	0
44	Nejla Bošnjaković	JU OŠ "Silvije Strahimir Kranjčević"	0	0	0	0	0	0
45	Mirza Šetkić	JU"Peta osnovna škola"Sokolovići	0	0	0	0	0	0
46	Nora Bešić	JU"Peta osnovna škola"Sokolovići	0	0	0	0	0	0

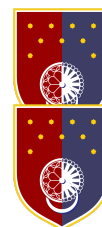


47	Harun Zahirović	OŠ "Đulistan" Lješevo, Ilijaš	0	0	0	0	0	0
48	Hamza Alihodžić	OŠ "Đulistan" Lješevo, Ilijaš	0	0	0	0	0	0
49	Merjem Fazlić	JU OŠ "Hamdija Kreševljaković" Sarajevo	0	0	0	0	0	0
50	Amar Tosuni	JU OŠ "Hamdija Kreševljaković" Sarajevo	0	0	0	0	0	0
51	Daris Tarić	Richmond Park Primary School Sarajevo	0	0	0	0	0	0



# Poredak škole

Redno mjesto	Naziv škole	Ukupno bodova
1	JU OŠ "Musa Ćazim Ćatić"	1067
2	JU OŠ „Čengić Vila 1“	484
3	JU Osman Nuri Hadžić	467
4	JU OŠ "Pofalići"	447
5	JU OŠ "ALEKSA ŠANTIĆ"	445
6	JU Osnovna škola "Grbavica II"	380
7	JU OŠ "Skender Kulenović"	380
8	Prva osnovna škola Ilidža	367
9	JU OŠ "9. maj" Pazarić, Sarajevo	355
10	Ćamil Sijarić	315
11	Richmond Park Primary School Sarajevo	275
12	JU "DESETA OSNOVNA ŠKOLA" ILIDŽA	270
13	Osnovna škola "Izet Šabić"	264
14	JU OŠ "Hamdija Kreševljaković" Sarajevo	145
15	JU"Peta osnovna škola"Sokolovići	59
16	OŠ "Đulistan" Lješevo, Ilijaš	18
17	JU OŠ "Silvije Strahimir Kranjčević"	10





# Postavke zadatka

## 1. Magični kvadrat

*Magični kvadrat* je kvadrat cijelih brojeva u kojem svi redovi, sve kolone i obje glavne dijagonale imaju jednak zbir brojeva. Ovaj zbir se naziva *magični zbir*. Primjer magičnog kvadrata je prikazan ispod.

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↓15
			↘15

Na ulazu će biti upisana 3 reda po 3 broja, koji predstavljaju kvadrat brojeva. Vaš zadatak je odrediti da li je upisani kvadrat magični, i ukoliko jeste, ispisati njegov magični zbir. Ako upisani kvadrat nije magični kvadrat, potrebno je ispisati broj -1.

### Ulazni podaci

U svakom od 3 reda se unose po 3 broja  $a_{i,j}$  koji predstavljaju elemente kvadrata brojeva. (**Napomena:** radi kompatibilnosti sa QBasic-om sistem za testiranje će unositi brojeve u zasebne redove, tako da će se ulaz zapravo sastojati od 9 redova po 1 broj. Prvo se unose brojevi prvog reda, pa drugog, pa trećeg.)



Ograničenja (program se neće testirati van ovih opsega)

$0 \leq a_{i,j} \leq 1000$ , pri čemu  $a_{i,j}$  predstavlja jedan element kvadrata.

Podzadatak 1 (18 bodova)

Uneseni kvadrat sigurno nije magičan.

Podzadatak 2 (31 bod)

Uneseni kvadrat sigurno jeste magičan.

Podzadatak 3 (51 bodova)

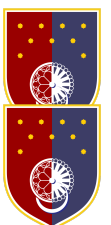
Bez dodatnih ograničenja.

### Izlazni podaci

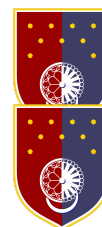
Ispisati magični zbir ukoliko je uneseni kvadrat brojeva, magični kvadrat. U suprotnom, ispisati -1.

### Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
2 7 6 9 5 1 4 3 8	15	Kvadrat je magični kvadrat i pri tome je magični zbir $2+7+6=15$ .
2 8 3 4 10 6	-1	Kvadrat nije magični kvadrat. Zbir dijagonala je $6+10+3 \neq 2+10+8$ .



6 7 8		
2 2 2 2 2 2 2 2 2	6	Kvadrat je magični kvadrat i pri tome je magični zbir $2+2+2=6$ .



## 2. Drvosječa

---

Pero drvosječa posjeduje ogromnu mašinu u koju se može ubaciti proizvoljan broj stabala. Nakon što se ubaci određen broj stabala, odredi se veličina reza  $R$  (u metrima). Veličina reza je prirodan broj, minimalno 2, ali ne smije biti veći ni od jednog stabla koje je ubačeno. Jednim klikom na dugme, mašina siječe svako od položenih stabala na jednake dijelove, svaki dio veličine  $R$ .

Međutim, mašina ima veliku manu - ukoliko neko ubačeno stablo nije moguće prerezati na jednake dijelove veličine  $R$ , mašina ne može izvršiti rezanje nijednog stabla dok se ne izbací pomenuto stablo. Obzirom da se Peri bliži kraj radnog dana, njega zanima koliko najviše stabala može ubaciti u mašinu, tako da je moguće da mašina uspješno izreže stabla u jednom potezu. Pogledati primjere za detaljnije obrazloženje.

**Napomena:** stablo veličine  $T$  može se izrezati rezom veličine  $T$  - iako taj rez neće uticati na veličinu stabla, stablo će nakon pritiska mašine poboljšati svoj kvalitet, tako da se to isplati u svakom slučaju.

### Ulazni podaci

U prvom redu se nalazi prirodan broj  $N$  koji predstavlja broj stabala koje Pero ima na raspolaganju. U sljedećih  $N$  redova se unosi  $N$  prirodnih brojeva koji predstavljaju visine stabala.

### Ograničenja

$$1 < N \leq 1000$$

Visina svakog stabla je prirodan broj između 1 i 1000 (uključujući 1 i 1000)

Veličina reza kojim se sijeku stabla mora biti prirodan broj strogo veći od 1.

Podzadatak 1 (10 bodova)

Visine svih stabala su parni brojevi.

Podzadatak 2 (15 bodova)

Visine svih stabala su prosti brojevi.

Podzadatak 3 (20 bodova)

Visine svih stabala su između 2 i 5.



#### Podzadatak 4 (55 bodova)

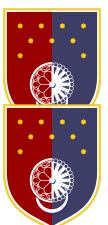
Bez dodatnih ograničenja.

#### Izlazni podaci

Program ispisuje 1 broj - najveći broj stabala koje Pero može staviti u mašinu, tako da ona u jednom potezu može odsjeći stabla na bilo koji validan način. Ranije je opisano kako i kada je moguće odrezati stabla.

#### Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
4 2 6 8 10	4	U mašinu možemo ubaciti sva stabla i podesiti veličinu reza na 2 metra. Prvo stablo će se isjeći na 1 komad veličine 2 metra, drugo na 3 komada po 2 metra, treće na 4 komada a peto na 5 komada. Dakle, moguće je u jednom potezu odsjeći sva 4 stabla.
3 4 7 12	2	Nije moguće ubaciti sva tri stabla u mašinu, obzirom da stablo veličine 7 nije nikako moguće odrezati na manje dijelove, osim ako je rez veličine 7, ali taj rez nije validan za ostala dva stabla.  Međutim, moguće je ubaciti prvo i zadnje stablo, a zatim sa rezom veličine 4 (ili 2, nije bitno) odsjeći stabla na manje dijelove.



3 3 5 7	1	<p>Nije moguće ubaciti 2 ili 3 stabla, te sve njih presjeći, obzirom da ne postoji veličina reza koja bi zadovoljila opisane uslove u zadatku.</p> <p>Najviše što mašina može uraditi u jednom potezu jeste odrezati jedno stablo, svejedno koje to stablo bilo.</p>
3 1 1 1	0	<p>Mašina ne može odrezati nijedno stablo, obzirom da su stabla veličine 1, a minimalan rez mašine je 2 metra, kao što je opisano u zadatku.</p>



### 3. Broj palindrom

---

Janja voli planine, piramide i palindrome. Želi da datom broju promijenite redoslijed cifara kako biste dobili najmanji mogući broj palindrom, ukoliko je to uopšte moguće uraditi.

Brojevi palindromi su cijeli brojevi koji se ne promijene obrtanjem redoslijeda cifara. Na primjer, brojevi 626 i 33 su brojevi palindromi, dok 755 i 4840 nisu. 010 nije validan oblik broja zato što počinje sa cifrom 0, tako da nije ni broj palindrom.

Vaš zadatak je da za uneseni broj  $N$ , odredite najmanji broj palindrom koji se dobija premješanjem cifara broja  $N$ , ako je to uopće moguće uraditi. Ukoliko nije moguće dobiti broj palindrom, ispisati "Nemoguće!" (bez navodnika).

#### Ulazni podaci

Ulaz se sastoji samo od jednog reda u kojem se unosi prirodan broj  $N$ .

Ograničenja (program se neće testirati van ovih opsega)

$$0 \leq N \leq 10^9$$

Podzadatak 1 (6 bodova)

$$0 \leq N < 10$$

Podzadatak 2 (20 bodova)

$$1 \leq N < 100$$

Podzadatak 3 (25 bodova)

Upisani broj ne sadrži cifru 0.

Podzadatak 4 (49 bodova)

Bez dodatnih ograničenja.

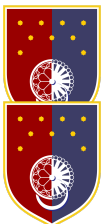


## Izlazni podaci

Ispisati najmanji broj palindrom ukoliko ga je moguće dobiti, u suprotnom ispisati "Nemoguće!".

## Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
122	212	Moguće je dobiti samo jedan broj palindrom, koji ujedno predstavlja i najmanji takav broj.
41481	14841	Svi mogući brojevi palindromi koji se dobijaju za $N = 41481$ su (14841, 41814). Od njih je najmanji 14841.
100	Nemoguće!	Jedini mogući palindrom je 010, koji nije validan broj po uslovima zadatka.





## 4. Mrežopolis

---

Mrežopolis je grad pun blokova zgrada između koji je gradska kvadratna mreža ulica. Za prvoaprilsku šalu, gradonačelnik Jednosmjernikus je učinio sve ulice jednosmjernim tako da iz svake raskrsnice morate otići u tačno jednom smjeru. Vaš prijatelj Odeon kreće u avanturu negdje u gradu, a zamolio je upravo vas da utvrdite da li će iz grada izaći ili će kružiti po gradu sve dok se pravila ne promijene. Uz to, trebate odrediti koliko će raskrsnica posjetiti na svojoj avanturi po Mrežopolisu (uključujući početnu).

Ukoliko će Odeon eventualno izaći iz grada potrebno je ispisati veliko slovo I, razmak, te broj različitih raskrsnica koje će posjetiti prije izlaska. Ukoliko će kružiti po gradu, potrebno je ispisati veliko slovo O, razmak, te (i u ovom slučaju) broj različitih raskrsnica koje će posjetiti na svojoj avanturi.

### Ulazni podaci

Prvi red upisa sadrži brojeve  $M$  i  $N$ , visina i širina grada na mapi. Raskrsnice su predstavljene u  $M$  redova i  $N$  kolona. Drugi red upisa sadrži brojeve  $K$  i  $L$ . Odeon kreće sa raskrsnice u redu  $K$  i koloni  $L$  na mapi. Sljedećih  $M$  redova sadrži po  $N$  karaktera "U", "D", "R" ili "L", koji označavaju da raskrsnica usmjerava stanovnike gore, dolje, desno ili lijevo na mapi, respektivno.

### Ograničenja (program se neće testirati van ovih opsega)

$$1 \leq M, N \leq 100,$$

$$1 \leq K \leq M, 1 \leq L \leq N.$$

Podzadatak 1 (12 bodova)

$$1 \leq M, N \leq 2$$

Podzadatak 2 (21 bod)

Sve raskrsnice su označene sa "U" ili "L", odnosno samo usmjeravaju stanovnike sjeverno ili zapadno.



Podzadatak 3 (28 bodova)

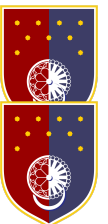
$$M = 1$$

Podzadatak 4 (39 bodova)

Bez dodatnih ograničenja.

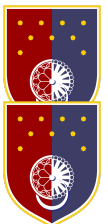
### Izlazni podaci

Prvi i jedini red ispisa treba sadržavati jedno od slova “O” ili “I”, u ovisnosti od toga da li Odeon kruži po gradu ili ne. Nakon tog slova je potrebno ispisati znak razmaka, te broj različitih raskrsnica koje posjeti na svojoj avanturi.



## Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
3 5 1 1 RDUDR URURL RRDRU	I 5	Odeon kreće iz gornjeg lijevog ćoška te se kreće redom <i>desno &gt; dolje &gt; desno &gt; gore &gt; gore</i> , nakon čega izlazi iz grada. Uključujući početnu, Odeon je posjetio 5 raskrsnica te izašao iz grada.
4 4 2 3 UUUU RDLD UDRR ULRU	O 7	Počevši na raskrsnici u drugom redu i trećoj koloni, Odeon se kreće <i>lijevo &gt; dolje &gt; dolje &gt; lijevo &gt; gore &gt; gore &gt; desno</i> , čime se vraća na istu raskrsnicu na kojoj je bio nakon prvog pomijeranja. Odeon će beskonačno kružiti po gradu, pri čemu će posjetiti 7 različitih raskrsnica.
3 3 3 1 RRD UUD ULL	O 8	Odeon kruži po rubu grada, čime će posjetiti ukupno 8 raskrsnica.



## 5. Isamovski brojevi

---

Isama interesuju brojevi, a posebno voli brojeve čije su susjedne cifre različite parnosti zbog svojih zanimljivih osobina i načina da se predvide. Ovi brojevi (u nastavku Isamovski brojevi) uključuju brojeve kao 145, 10583 i 7, a ne uključuju brojeve kao 33, 14792 i 228.

Za upisani broj  $N$  vaš zadatak je da njemu odredite najbliži Isamovski broj. Ukoliko postoje dva jednako udaljena Isamovska broja potrebno je da ih ispišete oba tako što ispišete manji pa veći, međusobno razdvojeni znakom razmaka.

### Ulazni podaci

Prvi i jedini red upisa sadrži broj  $N$ , broj kojem se treba odrediti najbliži Isamovski broj (ili brojevi).

Ograničenja (program se neće testirati van ovih opsega)

$1 \leq N < 10^{1000}$  (odnosno  $N$  je prirodan broj sa najviše 1000 cifara).

Podzadatak 1 (9 bodova)

$1 \leq N < 10$

Podzadatak 2 (12 bodova)

$1 \leq N < 100$

Podzadatak 3 (35 bodova)

$1 \leq N < 10^9$

Podzadatak 4 (13 bodova)

$N$  je za najviše 1 udaljen od Isamovskog broja.



## Podzadatak 5 (31 bodova)

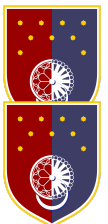
Bez dodatnih ograničenja.

### Izlazni podaci

Prvi i jedini red ispisa treba sadržavati jedan ili dva broja. To je najbliži Isamovski broj upisanom broju, odnosno najbliži brojevi ukoliko postoje dva Isamovska broja jednako bliska upisanom broju.

### Primjeri

Ulaz	Očekivani izlaz	Objašnjenje
5	5	Kako je 5 već Isamovski broj tako je on sam sebi najbliži (udaljenost 0).
55	54 56	Kako su i 54 i 56 Isamovski brojevi, a oba su za 1 udaljeni od 55 tako ih je oba potrebno ispisati.
157623	161010	161010 je najbliži Isamovski broj broju 157623. Ne postoji Isamovski broj koji počinje sa ciframa 1 5.
314159265358979 323846264338327 95028841971	309898989 898989898 989898989 898989898 98989	



# Rješenja zadataka

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

## 1. Magični kvadrat

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

using namespace std;

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

    vector<int> br_cifara(10);
    for(int i = 0; i < s.length(); i++)
        br_cifara[s[i]-'0']++;

    int br_neparnih = 0, indeks_neparne;

    for(int i = 0; i < 10; i++) {
        if(br_cifara[i] % 2) {
            br_neparnih++;
            indeks_neparne = i;
        }
    }

    int br_nenultih = s.length() - br_cifara[0];
    if(br_neparnih > 1 || (s.length() > 1 && br_nenultih < 2) ) {
        cout << "Nemoguće!";
        return 0;
    }

    string ispis;

    for(int i = 0; i < 10; i++)
        for(int j = 0; j < br_cifara[i]/2; j++)
```



```

        ispis += '0' + i;

    for(int i = 0; i < ispis.length(); i++) {
        if(ispis[i] != '0') {
            swap(ispis[i], ispis[0]);
            break;
        }
    }

    string kopija_ispisa = ispis;
    reverse(kopija_ispisa.begin(), kopija_ispisa.end());

    cout << ispis + (br_neparnih ? char('0' + indeks_neparne) + kopija_ispisa:
kopija_ispisa) << endl;
    return 0;
}

```



## 2. Semafor

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

using namespace std;

int main() {
    int broj_clanova;
    cin >> broj_clanova;

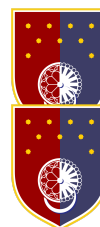
    vector<int> brojevi(broj_clanova);
    for (int i = 0; i < broj_clanova; i++)
        cin >> brojevi[i];

    int max_broj = 0;
    for (int gcd = 2; gcd < 1000; gcd++) {
        int broj_dijeljivih = 0;
        for (int i = 0; i < broj_clanova; i++)
            if (brojevi[i] % gcd == 0)
                broj_dijeljivih++;

        if (broj_dijeljivih > max_broj)
            max_broj = broj_dijeljivih;
    }

    cout << max_broj << endl;

    return 0;
}
```





### 3. Broj palindrom

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

using namespace std;

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

    vector<int> br_cifara(10);
    for(int i = 0; i < s.length(); i++)
        br_cifara[s[i] - '0']++;

    int br_neparnih = 0, indeks_neparne;

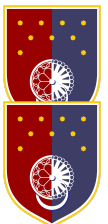
    for(int i = 0; i < 10; i++) {
        if(br_cifara[i] % 2) {
            br_neparnih++;
            indeks_neparne = i;
        }
    }

    int br_nenultih = s.length() - br_cifara[0];
    if(br_neparnih > 1 || (s.length() > 1 && br_nenultih < 2) ) {
        cout << "Nemoguće!";
        return 0;
    }

    string ispis;

    for(int i = 0; i < 10; i++)
        for(int j = 0; j < br_cifara[i]/2; j++)
            ispis += '0' + i;

    for(int i = 0; i < ispis.length(); i++) {
        if(ispis[i] != '0') {
            swap(ispis[i], ispis[0]);
            break;
        }
    }
}
```



```
string kopija_ispisa = ispis;
reverse(kopija_ispisa.begin(), kopija_ispisa.end());

cout << ispis + (br_neparnih ? char('0' + indeks_neparne) + kopija_ispisa:
kopija_ispisa) << endl;
return 0;
}
```



## 4. Mrežopolis

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

using namespace std;

int main() {
    int m, n, k, l;

    cin >> m >> n >> k >> l;
    k--;
    l--;

    vector<vector<char> > mapa(m, vector<char>(n));
    vector<vector<bool> > posjecena(m, vector<bool>(n));

    for(int i = 0; i < m; i++)
        for(int j = 0; j < n; j++)
            cin >> mapa[i][j];

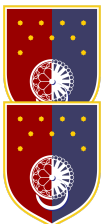
    int br = 0;
    bool izasao = false;

    while(!izasao && !posjecena[k][l]) {
        posjecena[k][l] = true;
        br++;

        if(mapa[k][l] == 'V') k++;
        else if(mapa[k][l] == '^') k--;
        else if(mapa[k][l] == '>') l++;
        else l--;

        if(k < 0 || k >= m || l < 0 || l >= n) izasao = true;
    }

    cout << (izasao ? "I " : "O ") << br << endl;
    return 0;
}
```



## 5. Isamovski brojevi

```
#include <iostream>

using namespace std;
bool parnost(int a){
    return a % 2;
}
void to_ascii(string &s) {
    for(int i = 0 ; i < s.length(); i++)
        s[i] += '0';
}

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

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

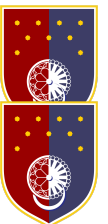
    int c = 0;
    for(int i = 1; i < s.length(); i++)
        if(parnost(s[i]) == parnost(s[i-1])) {
            c = i;
            break;
        }

    if(c == 0) {
        cout << s;
        return 0;
    }

    string s_veci = s, s_manji = s;
    s_veci[c]++;
    s_manji[c]--;

    for(int i = c + 1; i < s.length(); i++) {
        s_veci[i] = (parnost(s_veci[i-1]) ? 0 : 1);
        s_manji[i] = (parnost(s_manji[i-1]) ? 8 : 9);
    }

    if(s[c] == 0) {
        to_ascii(s_veci);
        cout << s_veci;
    }
}
```



```

        return 0;
    }

    if(s[c] == 9) {
        to_ascii(s_manji);
        cout << s_manji;
        return 0;
    }

    int blizi = -1, srednja_cifra;
    for(int i = c + 1; i < s.length(); i++) {
        if(parnost(s[c]) && parnost(i-c))
            srednja_cifra = 5;
        else if(!parnost(s[c]) && parnost(i-c))
            srednja_cifra = 5;
        else
            srednja_cifra = 4;

        if(s[i] < srednja_cifra) {
            blizi = 0;
            break;
        }

        if(s[i] > srednja_cifra) {
            blizi = 1;
            break;
        }
    }
    to_ascii(s_manji); to_ascii(s_veci);
    if(blizi == 0)
        cout << s_manji;
    else if(blizi == 1)
        cout << s_veci;
    else
        cout << s_manji << ' ' << s_veci;

    return 0;
}

```

