

### Zadatak 1 – Savršeni broj

Provjeriti da li je dati broj  $n$  savršen tj, jednak zbiru svih svojih pozitivnih djelilaca manjih od  $n$ . Ako nije štampati poruku „,  $n$  is not perfect.“ (npr. „32 is not perfect.“). Ako je broj  $n$  savršen, štampati broj kao zbir faktora, npr.  $6 = 1 + 2 + 3$ . **Ulaz:** U prvom i jedinom redu dat je prirodan broj  $n$  ( $2 \leq n \leq 2 \cdot 10^9$ ). **Izlaz:** Štampati ranije opisani tekst.

**Primjer:**

|   |   |
|---|---|
| <b>Ulaz</b><br>6<br><b>Izlaz</b><br>$6 = 1 + 2 + 3$ | <b>Ulaz</b><br>32<br><b>Izlaz</b><br>32 is not perfect. |
|---|---|

**Rješenje:** Koristi se klasa `ArrayList` za čuvanje djelilaca datog broja.

### Zadatak 2 – Obrnuta poljska notacija

U postfiksnoj notaciji (ili obrnutoj poljskoj notaciji, engl. reverse polish notation) operacija se zapisuje poslije dva operanda. Na primjer, zbir dva broja  $A$  i  $B$  zapisuje se kao  $A B +$ . Zapis  $B C + D *$  označava sljedeći izraz:  $(B + C) * D$ . Zapis  $A B C + D * +$  označava  $A + (B + C) * D$ . Prednost ovakvog načina zapisivanja je što nisu potrebne zagrade i nije potrebno da uvodimo pravila o redosljedu izvršavanja operacija. **Ulaz:** U prvom i jedinom redu dat je izraz u postfiksnoj notaciji koji sadrži samo jednocifrene prirdne brojeve i operacije  $+$ ,  $-$  i  $*$ . Simboli su razdvojeni sa po jednim blankom. **Izlaz:** Štampati vrijednost unesenog izraza.

**Primjer:**

|  |  |
|--|--|
| <b>Ulaz</b><br>8 9 + 1 7 - *<br><b>Izlaz</b><br>-102 | <b>Ulaz</b><br>8 9 +<br><b>Izlaz</b><br>17 |
|--|--|

**Rješenje:** Koristi se klasa `Stack`. Pokazati kako se karakter prevodi u broj koristeći `int i = zn - '0'`;

### Zadatak 3 – Pijane karte

U tzv. “igri pijanih” špil karata se podijeli na dvije jednake gomile, svakom igraču po jedna. Svaki od njih pokazuje po jednu kartu sa vrha špila i onaj čija je karta veća uzima karte i stavlja ih na dno svoje gomile. Onaj koji ostane bez karata gubi igru. Da bi uprostiti igru, smatraćemo da sve karte imaju različite vrjednosti i da najmanja karta pobjeđuje najveću. Igrač koji nosi karte, na dno svoje gomile prvo stavlja svoju kartu a zatim kartu protivnika (tj. karta protivnika je uvijek na dnu gomile). Napišite program koji modelira igru i štampa ko pobjeđuje i u koliko poteza. U igri učestvuje 10 karata numerisanih od 0 do 9; veća karta „nosi“ manju i karta 0 „nosi“ kartu 9. **Ulaz:** U prvom redu je 5 brojeva razdvojenih blankom – karte prvog igrača. U drugom redu je 5 brojeva razdvojenih blankom – karte drugog igrača. Karte su nabrojane od vrha prema dnu tj. igra počinje sa prvom kartom. **Izlaz:** U prvi i jedini red treba štampati riječ `first`, ako pobjeđuje prvi igrač ili `second` ako pobjeđuje drugi igrač. Zatim štampati jedan blanko i cio broj koji označava broj poteza do pobjede. Ako se igra ne završi za  $10^6$  poteza, štampati riječ `botva`.

**Primjer:**

| Ulaz                   | Izlaz    |
|------------------------|----------|
| 1 3 5 7 9<br>2 4 6 8 0 | second 5 |

**Rješenje:** Koristi se klasa `Queue`.

### Zadatak 4 – Skakač, izvor: Ruska informatička olimpijada

Date su dva polja šahovskoj tabli, početno (npr. A1) i završno (npr. H6). Napisati program koji određuje najmanji broj poteza koji je potreban skakaču da stigne sa početnog polja na krajnje polje. **Ulaz:** U prvom i jedinom redu unose se dva stringa razdvojena blankom. Svaki od stringova je dužine dva, a prvi karakter je jedno od slova A-H, dok je drugi karakter cifra 1-8. Smatrati da je ulaz ispravan. **Izlaz:** Štampati jedan cio broj – minimalni broj poteza potrebnih da se sa prvog učitano polja dođe do drugog učitano polja.

**Rješenje:** Koristi se klasa `Queue`. U red se stavljaaju pravovi (pozicija skakača, redni broj poteza)

**Zadatak 5 – Slikar (1 sec, 32 MB)**, izvor: HONI 2006, prvo kolo

Zli Kaktus čar dočepao se Čarobne bačvice i poplavio Čudesnu šumu! Slikar Paleta i tri mala ježa sada se moraju što brže vratiti do Dabrove kolibe, gdje će se skloniti od bujice.

Karta Čudesne šume se sastoji od R redova i S kolona. Na karti su prazna polja označena znakom '.', polja pokrivena vodom znakom '\*', a polja na kojima se nalazi stijena znakom 'X'. Takođe, dva su istaknuta polja, polje na kojem se nalazi Dabrova koliba (označeno znakom 'D') i polje na kojem se nalaze Slikar i tri mala ježa (označeno znakom 'S'). Slikar Paleta i tri mala ježa se u svakoj minuti mogu s polja na kojem se nalaze pomjeriti na neko od na 4 susjedna polja (gore, dolje, lijevo, desno). Svake minute se takođe i poplava proširi, i to tako da svako prazno polje koje je susjedno barem jednom već poplavljenom polju postaje poplavljeno. Ni voda, ni Slikar Paleta i tri mala ježa ne mogu proći kroz polje na kojem se nalazi stijena. Naravno, Slikar Paleta i tri mala ježa ne mogu proći kroz poplavljeno polje. Voda ne može poplaviti Dabrovu kolibu.

Napišite program koji će za unesenu kartu Čudesne šume ispisati **najkraće** vrijeme potrebno da Slikar Paleta i tri mala ježa sigurno dođu do Dabrove kolibe. **Napomena:** Slikar Paleta i tri mala ježa se ne smiju pomjeriti u polje koje će te minute biti poplavljeno. **Ulaz:** U prvom redu nalaziće se R i S, prirodni brojevi manji od ili jednaki 50. U sljedećih R redova nalaziće se po S znakova ('.', '\*', 'X', 'D' i 'S'). U cijeloj karti nalaziće se tačno jedan znak 'D' i tačno jedan znak 'S'. **Izlaz:** U jedan red treba štampati **najkraće** vrijeme potrebno da Slikar Paleta i tri mala ježa sigurno stignu do Dabrove kolibe. Ukoliko je to nemoguće potrebno je ispisati riječ "KAKTUS".

**Primjeri:**

|   |   |  |
|---|---|--|
| <b>ulaz</b><br>3 3<br>D.*<br>...<br>.S. | <b>ulaz</b><br>3 3<br>D.*<br>...<br>..S | <b>ulaz</b><br>3 6<br>D...*.<br>.X.X..<br>....S. |
| <b>izlaz</b><br>3                       | <b>izlaz</b><br>KAKTUS                  | <b>izlaz</b><br>6                                |

**Rješenje:** Slično prethodnom zadatku. Koristi se klasa Queue.

**Zadatak 6 – Infix u postfix**

Napisati program koji izraz u infiksnoj notaciji prevodi u odgovarajući postfixni izraz. **Ulaz:** U prvom i jedinom redu unosi string koji predstavlja infiksni izraz. String je dužine ne veće od 256 simbola, i sastoji se od pozitivnih cijelih brojeva i simbola '+', '-', '\*', '/', '(', ')'. **Izlaz:** Štampati jedan string – odgovarajući postfixni izraz, sa po jednim blankom između svaka dva simbola.

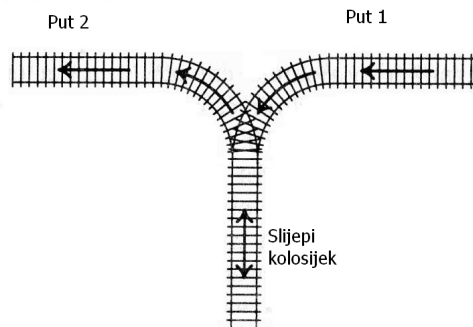
**Primjeri:**

|                               |  |  |
|-------------------------------|--|--|
| <b>ulaz</b><br>3+4*5/6        | <b>ulaz</b><br>(300+23) * (43-21) / (84+7) | <b>ulaz</b><br>(4+8) * (6-5) / ((3-2) * (2+2)) |
| <b>izlaz</b><br>3 4 5 * 6 / + | <b>izlaz</b><br>300 23 + 43 21 -* 84 7 + / | <b>izlaz</b><br>4 8 + 6 5 -* 3 2 -2 2 + * /    |

**Rješenje:** Koristi se klasa Stack.

**Zadatak 7 – Vozovi**

Ka slijepom kolosijeku dolazi voz sa strane označene kao Put 1 (vidi sliku).



Dozvoljeno je od voza otkočiti jedan ili više vagona (moguće i cio voz) sa prednje strane voza i uvesti ih na slijepi kolosijek. Poslije toga je moguće dio vagona sa slijepog kolosijeka izvesti na put 2. Zatim dio preostalih vagona sa Puta 1 ponovo možemo uvesti na slijepi kolosijek, pa dio vagona sa kolosijeka izvesti na Put 2, itd. Nije dozvoljeno pomjerati vagona sa Puta 2 na slijepi kolosijek niti izvoziti vagona sa kolosijeka na Put 1 (vidi strelice na slici). Poznato je kojim redom dolaze vagoni. Potrebno je, pomoću opisanih operacija, reorganizovati voz tako da vagoni idu u poretku 1,2,3.... Redosljed se posmatra na Putu 2, od slijepog kolosijeka. **Ulaz:** U prvom redu unosi se  $N$  – broj vagona u vozu ( $1 \leq N \leq 2000$ ). U drugom redu su dati vagoni koji dolaze po putu 1 ka slijepom kolosijeku. Vagoni su neumerisni brojevima od 1 do  $N$ . **Izlaz:** Štampati plan koji raspoređuje vagona u željeni raspored. Dopusnene su dvije operacije:

- 1  $K$  – pomjeranje  $K$  vagona sa puta 1 na slijepi kolosijek,  $K \geq 1$
- 2  $K$  – pomjeranje  $K$  vagona sa slijepog kolosijeka na put 2,  $K \geq 1$

Ako postoji više rješenja, štampati bilo koje od njih. Ako ne postoji rješenje, štampati samo broj 0.

**Primjeri:**

|                            |  |                           |
|----------------------------|--|---------------------------|
| <b>ulaz</b><br>3<br>3 2 1  | <b>ulaz</b><br>4<br>4 1 3 2              | <b>ulaz</b><br>3<br>2 3 1 |
| <b>izlaz</b><br>1 3<br>2 3 | <b>izlaz</b><br>1 2<br>2 1<br>1 2<br>2 3 | <b>izlaz</b><br>0         |

**Rješenje:** Koristi se klasa `Stack`.

**Zadatak 8 – Eratostenovo sito**

Primjenom Eratostenovog sita kreirati niz prostih brojeva ne većih od datog prirodnog broja  $n$ . **Ulaz:** U prvom i jedinom redu dat je prirodan broj  $n$  ( $1 \leq n \leq 2 \cdot 10^9$ ). **Izlaz:** Štampati dobijeni niz.

**Primjer:**

|                                     |
|-------------------------------------|
| <b>Ulaz</b><br>20                   |
| <b>Izlaz</b><br>2 3 5 7 11 13 17 19 |

**Rješenje:** Koristi se klasa `ArrayList` za cuvanje brojeva, jer ne znamo koliko ce biti prostih brojeva.