

# Prvi izpitni rok pri predmetu Programiranje 1

## 19. januar 2022

Oddajte datoteke Prva.java, Druga.java, Tretja.java in Cetrta.java. Testirate jih lahko takole:

(1) tj.exe Prva.java . . (2) tj.exe (3) tj.exe (4) tj.exe

- ① Praštevilo  $n$  je *Chenovo praštevilo*, če ima število  $n + 2$  največ štiri delitelje. Napišite program, ki prebere števili  $a \in [2, 1000]$  in  $b \in [a, 1000]$  in izpiše število Chenovih praštevil na intervalu  $[a, b]$ . (Lastnost morate preveriti *tudi* za število  $b$ : če je  $b$  praštevilo,  $b + 2$  pa ima največ štiri delitelje, potem je  $b$  Chenovo praštevilo na intervalu  $[a, b]$ .)

**Primer (vhod/izhod):**

40 50	2
-------	---

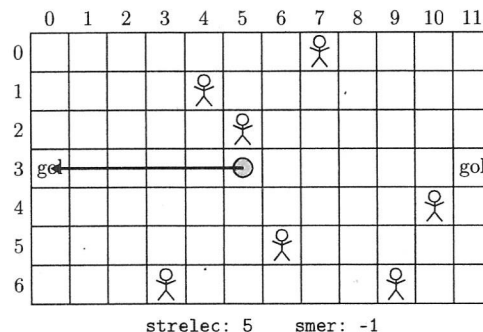
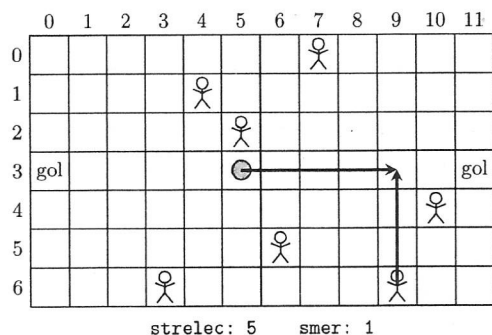
Chenovi praštevili sta 41 in 47.

- ② Na nogometnem igrišču velikosti  $h \times w$  ( $h \in [3, 999]$ ,  $w \in [3, 1000]$ ,  $h$  je lih) se gola nahajata na sredini leve in desne stranice (gl. sliko). Nogometaš z žogo stoji na sredini stolpca z indeksom  $s \in [1, w - 2]$  in v vodoravni črti strelja na levi ali desni gol. Po strelu se žoga vsako sekundo premakne za eno enoto (celico). Vsak igralec se v tem času premakne za eno enoto v smeri navzgor ali navzdol, lahko pa ostane pri miru. Vsak igralec poskuša žogo čimprej prestreči. Kolikšno razdaljo bo prepotovala žoga do prvega možnega prestrežanja?

V razredu Druga napišite metodo

```
public static int potZoge(int strelec, boolean[][] igralci, int smer),
```

ki vrne iskano razdaljo. Parameter `strelec` podaja začetni položaj žoge (število  $s$ ), elementi `true` v pravokotni tabeli `igralci` (velikosti  $h \times w$ ) podajajo položaje posameznih igralcev razen strelca, parameter `smer` pa podaja smer strela (1: desno; -1: levo). V primeru na levi sliki (Test01.java) je odgovor enak 4, v primeru na desni sliki (Test02.java) pa 5 (žoga zleti v gol, ker je nihče ne more prestreči):



Lahko predpostavite, da na levi in desni stranici ni nobenega igralca.

- ③ Geolov (angl. geocaching) je dejavnost, pri kateri igralci iščejo skrite »zaklade«. Ko igralec najde nek zaklad, se v dnevnik, ki pripada temu zakladu, vpiše ime najditelja in čas najdbe.

Podani so sledeči razredi (kot statični notranji razredi v razredu Tretja):

```
class Tocka {
    private double x; // koordinata x
    private double y; // koordinata y
}
class Najdba {
    private String najditelj; // ime najditelja
    private int cas; // čas najdbe (število dni od fiksne trenutka v preteklosti)
}
class Zaklad {
    private Tocka polozaj; // položaj zaklada
    private Najdba[] najdbe; // najdbe, urejene po naraščajočih časih
}
class Geolov {
    private Zaklad[] zakladi; // vsi zakladi
}
```

V razredu Tretja napišite sledeče metode:

- [32%] `public String[] najditelji()` v razredu Zaklad:  
Vrne tabelo imen vseh najditeljev zaklada `this` (*i*-ti element vrnjene tabele naj bo ime najditelja za *i*-to najdbo).
- [34%] `public double pot(Tocka zacetna)` v razredu Geolov:  
Vrne skupno dolžino poti, ki jo opravimo, če pričnemo v podani začetni točki, po vrsti obiščemo vse zaklade (tako, kot so nanizani v tabeli `this.zakladi`) in se vrnemo v začetno točko.
- [34%] `public Zaklad nazadnjeOdkrit()` v razredu Geolov:  
Vrne zaklad z najpoznejšim časom prve najdbe oziroma `null`, če igralci niso našli še nobenega zaklada.

- ④ V razredu Cetrti napišite sledeči metodi:

- [50%] `public static <T> List<T> odsek(Iterator<T> it, int p, int q)`  
Vrne seznam tistih členov zaporedja, določenega s podanim iteratorjem, ki imajo indekse med vključno *p* in vključno *q*. Lahko predpostavite, da je zaporedje dolgo najmanj *q*. (Kot običajno ima prvi člen zaporedja indeks 0.)
- [50%] `public static Iterator<Integer> manjkajoci(Iterator<Integer> it, int zacetek)`

Iterator *it* se sprehaja po neskončnem zaporedju zaporednih števil tipa `int` (število `Integer.MAX_VALUE` sledi `Integer.MIN_VALUE`), ki se prične s številom *zacetek*, vendar nekatera števila izpusti. Metoda `manjkajoci` naj vrne iterator, ki se sprehaja po zaporedju števil, ki jih iterator *it* izpusti. Na primer, če je *zacetek* enak 42, prvih pet členov zaporedja, ki ga določa iterator *it*, pa 43, 44, 46, 49 in 53, naj bo prvih sedem členov zaporedja, ki ga določa vrnjeni iterator, enakih 42, 45, 47, 48, 50, 51 in 52.

Metoda `hasNext` vrnjenega iteratorja naj vedno vrne `true`.