

R smer, Dizajn programskih jezika, grupa A, 9.2.2017.

Na *Desktop*-u napraviti direktorijum čije je ime u formatu **dpj_j2_ImePrezime_BrojIndeksa**.

Na primer, **dpj_j2_JovanPetrovic_mr14072**. Sve zadatke sačuvati u ovom direktorijumu.

NAPOMENA: 1. i 2. zadatak se rađa u programskom jeziku Java, a 3. i 4. u programskom jeziku C++. Eliminacioni test primeri su obeleženi zvezdicom. Potrebno je da se strogo držite formata ispisa koji je naznačen u zadacima. Da biste osvojili poene, neophodno je da se program uspesno kompilira.

1. (20%) Korisnik unosi dimenziju niza, a zatim i njegove elemente, prirodne petocifrene brojeve, sa standardnog ulaza. Napisati program koji proverava koji su elementi niza palindromi i ispisuje ih na standardni izlaz. Za proveru da li je element niza palindrom i njegov ispis u tom slučaju, koristiti zasebne niti. Broj niti treba da odgovara broju procesora koji su na raspolaganju.

UPUTSTVO: Broj je palindrom ako se isto čita sa obe strane, na primer 12321.

NAPOMENA: Pretpostaviti da su ulazni podaci ispravni. Redosled ispisa ne mora biti isti kao u test primerima.

Primer 1*	Primer 2	Primer 3	Primer 4
<pre>ULAZ: 3 10001 20202 12345 IzLAZ: 10001 20202</pre>	<pre>ULAZ: 4 10202 10201 54321 54345 IzLAZ: 10201 54345</pre>	<pre>ULAZ: 3 78787 98765 45655 IzLAZ: 78787</pre>	<pre>ULAZ: 4 10434 45700 10011 70808 IzLAZ:</pre>

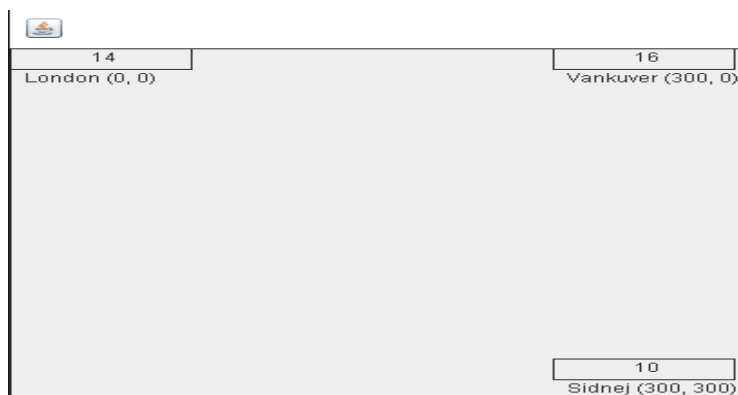
2. (30%) Kontroloru letenja je potrebna pomoć oko odvijanja vazdušnog saobraćaja na tri internacionalna aerodroma u Londonu, Vankuveru i Sidneju. Napisati program koji simulira letove između ova tri aerodroma. Sa svakog aerodroma poleću avioni, a kada svi avioni obiđu sve aerodrome program se zaustavlja.

Aerodromi:

Internacionalni aerodromi su raspoređeni na panelu dimenzije 500×500 . Sa svakog aerodroma poleću avioni, pri čemu je inicijalno stanje broja aviona na aerodromu u Londonu 14, Vankuveru 16 i Sidneju 10. Tačne pozicije aerodroma date su na slici ispod. Broj aviona na nekom aerodromu je deljeni resurs jer u nekom momentu neki avioni mogu da sleću i time povećavaju broj aviona, a drugi da poleću i tako smanjuju broj aviona na aerodromu. Pri kreiranju aerodroma konstruktoru se prosleđuju grad u kom je aerodrom (London, Vankuver ili Sidnej), koordinate aerodroma i inicijalno stanje broja aviona. Aerodrome grafički predstaviti kao pravougaonike dimenzije 100×20 . Na sredini pravougaonika ispisati trenutno stanje broja aviona, a ispod pravougaonika grad u kom se nalazi aerodrom, kao i tačne koordinate aerodroma (videti na slici ispod).

Avioni:

Algoritam kretanja aviona je takav da ako su na aerodromu, onda miruju 2 sekunde, a nakon toga na nasumičan način biraju drugi aerodrom gde će da krenu i počinju da lete ka njemu pravolinijski. Avion se pomera sa jediničnim vektorom pomeraja na svakih 10 milisekundi. Kada avion dođe u zonu aerodroma, on sleti na njega nakon čega ponovo miruje 2 sekunde pa zatim nastavlja dalje. Jedan avion može obići iste aerodrome više puta. Kada avion obiđe sva tri aerodroma, staje. Svaki avion treba implementirati kao zasebnu nit i grafički ga predstaviti popunjenim krugom prečnika 10. Pri kreiranju ovih objekata konstruktoru se prosleđuju identifikator aviona (ceo pozitivan broj koji označava redni broj aviona, računajući sve aerodrome zajedno: 1, 2, 3, ...) i koordinate aerodroma na kome se avion na početku nalazi.



Slika 1: Na slici je dat grafički prikaz aerodroma. U pravougaonicima je prikazano inicijalno stanje broja aviona, a ispod njih naziv mesta u kome je aerodrom i tačne koordinate aerodroma.

NAPOMENA: Ovak zadatak možete rešiti bez pravljenja grafičkog interfejsa. U tom slučaju možete dobiti maksimalno $\frac{2}{3}$ od broja poena na ovom zadatku. U ovoj verziji rešenja, potrebno je ispisivati na standardni izlaz kada određeni avion poleće ili sleće na neki aerodrom. Na primer:

Avion 3 je sleteo na aerodrom u Londonu.

Avion 10 je poleteo sa aerodroma u Sidneju.

3. (20%) Napisati šablon klasu Kvadar u okviru prostora imena *geometrija* koja kao polja podatke ima širinu, dužinu i visinu kvadra. Sva tri podatka su tipa *T* koji je parametar šablona. U klasi obezbediti:

- konstruktor sa tri argumenta
- metod za računanje površine kvadra (pretpostaviti da za objekte tipa *T* postoje definisane operacije množenja i sabiranja i da će rezultat izračunavanja biti tipa *T*)
- operatore poređenja *>*, *==* (poređenje kvadara se svodi na poređenje njihovih površina, pretpostaviti da za objekte tipa *T* postoje definisani operatori poređenja)

Napisati program koji testira šablon klasu Kvadar za podatke tipa *int*. Program prvo učitava podatke za dva kvadra sa celobrojnim poljima i ispisuje 0 ako nisu jednaki i površinu za veći od unetih kvadara ili 1 ako su jednaki i površinu bilo kog kvadra. Pretpostaviti da su ulazni podaci ispravni.

Primer 1*

```
ULAZ:
 2 6 10
 11 3 20
IZLAZ:
0
626
```

Primer 2

```
ULAZ:
 3 3 3
 2 2 2
IZLAZ:
0
54
```

Primer 3

```
ULAZ:
 2 6 10
 2 6 10
IZLAZ:
1
184
```

Primer 4

```
ULAZ:
 2 2 2
 2 2 2
IZLAZ:
1
24
```

4. (30%) Nakon održane *Super lige Srbije* potrebno je odrediti utakmice u kojima je postignuta određena gol razlika. Napisati program koji pomaže u određivanju utakmica na osnovu podataka o fudbalskim duelima, pri čemu je kriterijum izbora tražena gol razlika.

- Napisati klasu pod nazivom *Duel* koja sadrži podatke o odigranoj utakmici. Klasa kao polja podatke ima nazive dva kluba koji su odigrali duel i broj golova koji je postigao svaki klub.
- Napisati funkcional pod nazivom *Gol* koji za argument prihvata vrednost koja predstavlja traženu gol razliku i služi za proveru da li neki duel zadovoljava uslov ili ne. Ukoliko vrednost prosledena konstruktoru nije ispravna, podrazumevano postaviti da je tražena gol razlika 2.
- Napisati šablon funkciju koja za argumente dobija kolekciju objekata klase *Duel* i proizvoljan funkcional i ispisuje imena klubova koji su odigrali utakmicu sa traženom gol razlikom ili odgovarajuću poruku ukoliko takvih duela nije bilo.
- Sa standardnog ulaza se unosi ukupan broj odigranih duela, podaci o njima i tražena gol razlika. Podaci o duelu se unose u sledećem redosledu *naziv kluba domaćina*, *naziv gostujućeg kluba*, *broj postignutih golova domaćina*, *broj postignutih golova gostujućeg kluba*. Pretpostaviti da su svi podaci o utakmicama ispravno zadati i da će naziv kluba uvek biti jedna reč. Na standardni izlaz ispisati imena klubova koji su odigrali utakmicu sa traženom gol razlikom.

Primer 1*

```
ULAZ:
3
Radnicki Metalac 0 2
Partizan Spartak 4 1
Vojvodina Radnik 3 0
2
IZLAZ:
Radnicki Metalac
```

Primer 2

```
ULAZ:
3
Radnicki Partizan 2 5
Borac Spartak 3 0
Vojvodina Metalac 0 0
3
IZLAZ:
Radnicki Partizan
Borac Spartak
```

Primer 3

```
ULAZ:
2
Radnicki Metalac 0 2
Partizan Spartak 4 1
-4
IZLAZ:
Radnicki Metalac
```

Primer 4

```
ULAZ:
4
Borac Spartak 3 0
Vojvodina Metalac 0 0
Partizan Radnicki 1 0
Napredak Cukaricki 1 1
-3
IZLAZ:
Nema trazениh duela.
```