

R smer, Dizajn programskih jezika, Septembar 1, 14.9.2017.

Na *Desktop*-u napraviti direktorijum čije je ime u formatu **dpj_sep1_ImePrezime_BrojIndeksa**.

Na primer, **dpj_sep1_JovanPetrovic_mr14072**. Sve zadatke sačuvati u ovom direktorijumu.

NAPOMENA: 1. i 2. zadatak se rade 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 uspešno kompilira.

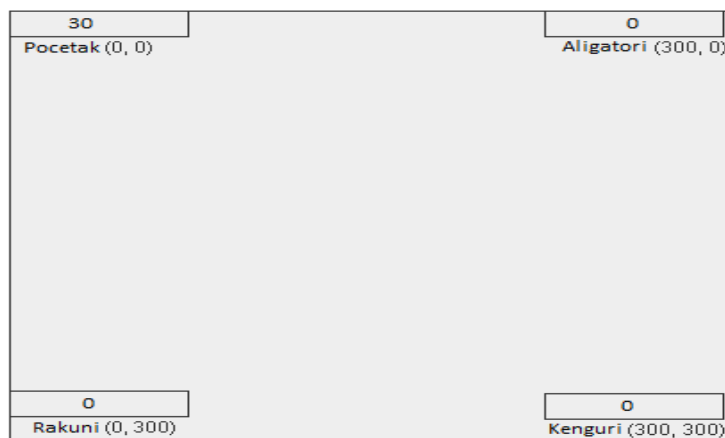
1. (20%) Korisnik prvo unosi dimenziju niza, a zatim i njegove elemente, cele četvorocifrene brojeve, sa standardnog ulaza. Napisati program koji za svaki element niza određuje najveći broj koji se može zapisati pomoću njegovih cifara i ispisuje ga na standardni izlaz. Na primer, najveći broj koji se može zapisati pomoću cifara broja 1070 je 7100. Za određivanje traženih brojeva i njihov ispis, koristiti zasebne niti. Broj niti treba da odgovara broju procesora koji su na raspolaganju.
- 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 1234 2254 7878 IZLAZ: 4321 5422 8877</pre>	<pre>ULAZ: 2 -1020 2200 IZLAZ: 2100 2200</pre>	<pre>ULAZ: 3 -1001 -1100 -1010 IZLAZ: 1100 1100 1100</pre>	<pre>ULAZ: 2 1111 -1111 IZLAZ: 1111 1111</pre>

2. (30%) Odeljenje učenika osnovne škole "Milica Pavlović" danas ide u posetu Beo zoo vrtu. Učiteljica im je rekla da obavezno posete najstarijeg aligatora na svetu, kenguricu sa mladunčetom, kao i zanimljive rakune i da se na kraju obilaska vrate na mesto polaska. Napisati program koji simulira kretanje đaka u obilasku zoo vrta čija je mapa sa staništima životinja prikazana na slici ispod. Đaci kreću u obilazak sa lokacije *Početak*, a kada svi đaci obiđu sve životinje i vrate se na lokaciju *Početak* program se zaustavlja.

Lokacije: Postoje dve vrste lokacija u zoo vrtu - početna lokacija i veštačka staništa životinja i sve su raspoređene na panelu dimenzije 500×500 . Tačne pozicije date su na slici ispod. Broj đaka na nekoj lokaciji je deljeni resurs jer u jednom momentu neki đaci stižu na lokaciju i time povećavaju ukupan broj đaka koji se tu trenutno nalazi, a drugi napuštaju lokaciju i tako smanjuju broj đaka na lokaciji. Pri kreiranju lokacija konstruktoru se prosleđuje naziv lokacije (*Početak* ukoliko je u pitanju početna lokacija ili naziv vrste životinja koje tu žive) i tačne koordinate. Lokacije grafički predstaviti kao pravougaonike dimenzije 100×20 . Na sredini pravougaonika ispisati trenutno stanje broja đaka, a ispod pravougaonika vrstu životinje koja tu živi, kao i tačne koordinate lokacije (videti na slici ispod).

Đaci: Svaki đak kreće u obilazak sa lokacije *Početak*, na kojoj bude 4 sekunde nakon čega na nasumičan način bira prvo stanište koje želi da poseti i počinje da se kreće pravolinijski ka tom staništu (smatrati da između svake dve lokacije postoji direktan pravolinijski put koji ih povezuje). Đak se pomera sa jediničnim vektorom pomeraja na svakih 10 milisekundi. Kada dođe u zonu sledećeg staništa, ponovo miruje 4 sekunde pa zatim nastavlja dalje svoj obilazak životinja. Jedan đak može obići iste vrste životinja više puta. Kada đak obiđe sva tri staništa, najkraćim putem se vraća na lokaciju *Početak* i tu se zaustavlja. Svakog đaka treba implementirati kao zasebnu nit i grafički ga predstaviti popunjenim krugom prečnika 10. Pri kreiranju ovih objekata konstruktoru se prosleđuje identifikator (ceo pozitivan broj koji označava redni broj đaka u dnevniku, računajući sve đake zajedno: 1, 2, 3, ...). Odeljenje ima 30-oro đaka i svi su pošli u zoo vrt.



Slika 1: Na slici je data mapa Beo zoo vrta. U pravougaonicima je prikazano inicijalno stanje broja đaka na svakoj lokaciji na početku, a ispod njih naziv i tačne koordinate lokacije.

NAPOMENA: 2. 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 dak dolazi ili odlazi sa neke lokacije. Na primer:

Dak sa rednim brojem 3 je stigao na stanište aligatora.

Dak sa rednim brojem 10 je krenuo sa staništa kengura.

3. (20%) Napisati šablon klasu *Radnik* u okviru prostora imena *kompanija*. Radnika karakterišu jedinstveni identifikacioni broj *id*, tipa *int* i njegova plata (u dinarima), tipa *T* koji je parametar šablona. U klasi obezbediti:

- konstruktor sa dva argumenta za polja podatke,
- operatore poređenja *<*, *==* (poređenje radnika se svodi na poređenje njihovih plata, pretpostaviti da za objekte tipa *T* postoje definisani operatori poređenja),
- metode za računanje visine plate izražene u evrima i dolarima (smatrati da je kurs evra 119.2, a dolara 100.3), pretpostaviti da za objekte tipa *T* postoji definisano množenje sa konstantama i da će rezultat izračunavanja biti tipa *T*.

Napisati program koji testira šablon klasu *Radnik* za platu tipa *double*. Program prvo učitava podatke za dva radnika sa realnim platama (identifikacioni broj i platu radnika redom), poredi ih i ukoliko su jednaki, ispisuje njihovu platu (bez dupliranja) izraženu u dolarima, a ukoliko nisu, ispisuje identifikacioni broj i platu većeg izraženu u evrima. U slučaju neispravnih ulaznih podataka ispisati -1 i prekinuti program.

Primer 1*

```
ULAZ:
123
54000.00
124
53000.00
IZLAZ:
123
453.02
```

Primer 2

```
ULAZ:
123
63000.00
124
63000.00
IZLAZ:
628.116
```

Primer 3

```
ULAZ:
115
102000.00
100
165000.00
IZLAZ:
100
1384.23
```

Primer 4

```
ULAZ:
101
97000.00
-102
97000.00
IZLAZ:
-1
```

4. (30%) Jovan želi da obraduje baku i da joj kupi jedan poklon u radnji. On na raspolaganju ima određenu količinu novca i zanima ga koji to artikal može da priušti. Napisati program koji pomaže Jovanu u kupovini poklona na osnovu podataka o artiklima (naziva i njihovih cena) pri čemu je kriterijum izbora količina novca kojom Jovan raspolaže.

- Napisati funkcional pod nazivom *PovoljanArtikal* koji za argument prihvata vrednost koja predstavlja količinu novca kojom kupac raspolaže i služi za proveru da li neki artikal može biti kupljen tim novcem ili ne. Ukoliko vrednost prosledena konstruktoru nije ispravna, podrazumevano postaviti količinu novca na 1000.
- Napisati šablon funkciju koja za argumente dobija kolekciju sa podacima o artiklima i proizvoljan funkcional i ispisuje nazive povoljnih artikala, tj. artikala koji mogu da se priušte za datu količinu novca ili odgovarajuću poruku ukoliko takvih nema.
- Sa standardnog ulaza se unosi ukupan broj artikala (ceo pozitivan broj), a zatim podaci o artiklima i količina novca kojom Jovan raspolaže (ceo pozitivan broj). Podaci o artiklima sadrže *Naziv* (pretpostaviti da će biti jedinstvena niska karaktera) i *Cenu* svakog artikla (realan broj). Pretpostaviti da su svi podaci ispravno zadati. Na standardni izlaz ispisati:

1) učitane podatke o artiklima u obliku *Naziv Cena* sortirano rastuće po ulaznom podatku *Naziv*.

2) nazive artikala koje Jovan može da kupi (sortirano).

Primer 1*

```
ULAZ:
3
ogledalo 999.00
snala 450.50
marama 1500.00
500
IZLAZ:
marama 1500.00
ogledalo 999.00
snala 450.50
snala
```

Primer 2

```
ULAZ:
3
penkalo 780.50
sesir 2500.00
narukvica 1000.00
1000
IZLAZ:
narukvica 1000.00
penkalo 780.50
sesir 2500.00
narukvica
penkalo
```

Primer 3

```
ULAZ:
2
sesir 1500.00
ogrlica 2500.00
900
IZLAZ:
ogrlica 2500.00
sesir 1500.00
Nema povoljnih artikala.
```

Primer 4

```
ULAZ:
2
sesir 500.00
ogrlica 2500.00
-100
IZLAZ:
ogrlica 2500.00
sesir 500.00
sesir
```