

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Rezultati i uvidi u kolokvije:** Rezultati u utorak, 28.6., navečer na webu, a uvidi u srijedu, 29.6., u 11 sati.

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

**Zadatak 1.** (10 + 10 = 20 bodova)

U predavaonici se nalazi  $n$  osoba koje imaju ime, prezime (stringovi od najviše 50 znakova), spol (znak), broj godina, te osobni identifikacijski broj (OIB). Osobe mogu biti u predavaonici s određenim brojem svojih prijatelja.

- (a) Konstruirajte strukturu `osoba` koja omogućava spremanje informacija o određenoj osobi i njezinim prijateljima. Napišite funkciju prototipa `void brisi(osoba* predavaonica, int n)` koja, iz polja `predavaonica` i iz svih struktura u njemu, briše sve one osobe koje imaju (strogo) **više od tri muška** prijatelja.

Rješenje u kojem je za pronalazak prijatelja određene osobe potrebno ispitati sve osobe iz predavaonice nosi maksimalno 7 bodova.

- (b) Ispišite sve parove osoba, u obliku `(Ime_prve, Prezime_prve - Ime_druge, Prezime_druge)`, koje imaju **barem 80%** zajedničkih prijatelja.

Zajednički prijatelji dvije osobe  $o_1$  i  $o_2$  su osobe različite od  $o_1, o_2$ , koje poznaju i osobu  $o_1$  i osobu  $o_2$ .

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 2.** (20 bodova) Napišite funkciju prototipa `char* ponavljanje(char** popis, int n)` koja vraća pokazivač na string, izgrađen od praznog stringa na sljedeći način. Varijabla `popis` pokazuje na već postojeći niz u kojem je zapisano  $n$  riječi. Korisnik upisuje riječi (po jednu u redu), sve dok ne upiše znak '#'. Za svaku učitane riječ  $r$ , funkcija `ponavljanje` pretražuje zadani niz od  $n$  riječi i traži prvu riječ  $w$ , takvu da se riječ  $r$  može dobiti konačnim ponavljanjem riječi  $w$ . Ako nađe takvu riječ  $w$ , onda tu riječ treba dodati na kraj stringa koji se izgrađuje, tako da je među dodanim riječima po jedan razmak. U suprotnom, string se ne mijenja. Smijete pretpostaviti da učitane riječi nemaju više od 80 znakova.

Primjer: riječ  $r = \text{jojjojoj}$  dobivena je trostrukim ponavljanjem riječi  $w = \text{joj}$ .

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 3.** (2 + 6 + 7 = 15 bodova) Elementi vezane liste trebaju sadržavati podatke o dioničarima jedne tvrtke, i to: ime (s najviše 30 znakova), te broj dionica (prirodni broj).

- (a) Napišite definiciju tipa podatka za pojedini element liste, na način da bude moguće definirati varijable naredbom `dionicar item;` i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `dionicar* trgovina(dionicar* first, char* ime1, char* ime2)`, gdje je `first` pokazivač na prvi element liste, a `ime1` je ime dioničara koji otkupljuje dionice od dioničara `ime2`. Kupovina se treba obaviti na način da se dioničaru `ime1` pribroje sve dionice dioničara `ime2`, te da se obriše element vezane liste u kojem su spremljeni podaci dioničara `ime2`. Ukoliko barem jedan dioničar zadanog imena ne postoji u vezanoj listi, funkcija ne radi nikakvu promjenu u ulaznoj vezanoj listi. Funkcija vraća pokazivač na početak vezane liste.
- (c) Napišite funkciju `dionicar* najbogatiji(dionicar** first)` koja iz liste izdvaja najboljih 5% dioničara, na način da se formira nova vezana lista, u kojoj su ti elementi spremljeni (sortirani silazno po broju dionica), i vraća se pokazivač na početak nove vezane liste. `first` je pokazivač na početak (stare) vezane liste. Broj najboljih 5% dioničara odredi se tako da se uzme najveći cijeli broj dioničara manji ili jednak od 5% ukupnog broja dioničara, ukoliko je taj broj veći od 1, odnosno, 1, ukoliko je taj broj jednak 0. Ukoliko je ulazna lista prazna, funkcija ne radi nikakvu promjenu na ulaznoj vezanoj listi, a vraća `NULL` pokazivač. Ovdje nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima vezane liste.

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 4.** (20 bodova) Deklarirajte strukturu `ispit` koja se sastoji (samo) od sljedećih stavki: ime i prezime studenta (string do 50 znakova), ime kolegija (string do 100 znakova) te ocjena (cijeli broj). Niz ovakvih struktura čuva se u binarnoj datoteci `ispiti.dat` i on predstavlja ocjene koje su studenti ostvarili na ispitima.

Napišite program koji, za svakog studenta, računa njegov prosjek ocjena i ispisuje sve studente i njihove prosjeke u tekstualnu datoteku `studenti.txt`, tako da ih sortira silazno po prosjecima. Ispisuje se jedan student po liniji, u obliku “prezime ime: prosjek” (bez navodnika), a prosjek je zaokružen na tri decimale.

Smijete pretpostaviti da će biti najviše 1000 studenata. Smijete koristiti jednu dodatnu strukturu i jedan dodatni niz.

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Rezultati i uvidi u kolokvije:** Rezultati u utorak, 28.6., navečer na webu, a uvidi u srijedu, 29.6., u 11 sati.

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

**Zadatak 1.** (10 + 10 = 20 bodova)

U kuglani se nalazi  $n$  osoba koje imaju ime, prezime (stringovi od najviše 50 znakova), spol (znak), broj godina, te osobni identifikacijski broj (OIB). Osobe mogu biti u kuglani s određenim brojem svojih prijatelja.

- (a) Konstruirajte strukturu osoba koja omogućava spremanje informacija o određenoj osobi i njezinim prijateljima. Napišite funkciju prototipa `void brisi(osoba* kuglana, int n)` koja, iz polja `kuglana` i iz svih struktura u njemu, briše sve one osobe koje imaju (strogo) **manje od dva ženska** prijatelja.

Rješenje u kojem je za pronalazak prijatelja određene osobe potrebno ispitati sve osobe iz kuglane nosi maksimalno 7 bodova.

- (b) Ispišite sve parove osoba, u obliku (`Ime_prve, Prezime_prve - Ime_druge, Prezime_druge`), koje **nemaju niti jednog** zajedničkog prijatelja.

Zajednički prijatelji dvije osobe  $o_1$  i  $o_2$  su osobe različite od  $o_1, o_2$ , koje poznaju i osobu  $o_1$  i osobu  $o_2$ .

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

### Zadatak 2. (20 bodova)

Napišite funkciju prototipa `char* podniz(char** lista, int m)` koja vraća pokazivač na string, izgrađen od praznog stringa na sljedeći način. Varijabla `lista` pokazuje na već postojeći niz u kojem je zapisano  $m$  riječi. Korisnik upisuje riječi (po jednu u redu), sve dok ne upiše znak '?'. Za svaku učitane riječ  $a$ , funkcija `podniz` pretražuje zadani niz od  $m$  riječi i traži prvu riječ  $b$ , koja sadrži riječ  $a$ . Ako nađe takvu riječ  $b$ , onda tu riječ treba dodati na kraj stringa koji se izgrađuje, tako da je među dodanim riječima po jedan razmak. U suprotnom, string se ne mijenja. Smijete pretpostaviti da učitane riječi nemaju više od 80 znakova.

Primjer: riječ  $a = \text{dan}$  sadržana je u riječima  $b = \text{danas, predan, predanost}$ .

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 3.** ( $2 + 6 + 7 = 15$  bodova) Elementi vezane liste trebaju sadržavati podatke o djeci jednog naselja, i to: ime (s najviše 25 znakova), te broj sličica nogometnih igrača koje su prikupili (prirodni broj).

- (a) Napišite definiciju tipa podatka za pojedini element liste, na način da bude moguće definirati varijable naredbom `djeca item`; i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `djeca* igra(djeca* first, char* dijete1, char* dijete2)`, gdje je `first` pokazivač na prvi element liste, a `dijete1` je ime djeteta koje je u igri dobilo sve sličice djeteta `dijete2`. Igra se treba obaviti na način da se djetetu `dijete1` pribroje sve sličice `dijete2`, te da se obriše element vezane liste u kojem su spremljeni podaci djeteta `dijete2`. Ukoliko barem jedno dijete zadanog imena ne postoji u vezanoj listi, funkcija ne radi nikakvu promjenu u ulaznoj vezanoj listi. Funkcija vraća pokazivač na početak vezane liste.
- (c) Napišite funkciju `djeca* najbolji(djeca** first)` koja iz liste izdvaja najboljih 10% djece, na način da se formira nova vezana lista, u kojoj su ti elementi spremljeni (sortirani silazno po broju sličica), i vraća se pokazivač na početak nove vezane liste. `first` je pokazivač na početak (stare) vezane liste. Broj najboljih 10% djece odredi se tako da se uzme najveći cijeli broj djece manji ili jednak od 10% ukupnog broja djece, ukoliko je taj broj veći od 1, odnosno, 1, ukoliko je taj broj jednak 0. Ukoliko je ulazna lista prazna, funkcija ne radi nikakvu promjenu na ulaznoj vezanoj listi, a vraća `NULL` pokazivač. Ovdje nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima vezane liste.

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 4.** (20 bodova) Deklarirajte strukturu `ispit` koja se sastoji (samo) od sljedećih stavki: ime i prezime studenta (string do 50 znakova), ime kolegija (string do 100 znakova) te ocjena (cijeli broj). Niz ovakvih struktura čuva se u binarnoj datoteci `ispiti.dat` i on predstavlja ocjene koje su studenti ostvarili na ispitima.

Napišite program koji, za svaki kolegij, računa prosjek ocjena i ispisuje sve kolegije i njihove prosjeke u tekstualnu datoteku `kolegiji.txt`, tako da ih sortira silazno po prosjecima. Ispisuje se jedan kolegij po liniji, u obliku “`ime: prosjek`” (bez navodnika), a prosjek je zaokružen na tri decimale.

Smijete pretpostaviti da će biti najviše 100 kolegija. Smijete koristiti jednu dodatnu strukturu i jedan dodatni niz.



## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Rezultati i uvidi u kolokvije:** Rezultati u utorak, 28.6., navečer na webu, a uvidi u srijedu, 29.6., u 11 sati.

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

**Zadatak 1.** (10 + 10 = 20 bodova)

U restoranu se nalazi  $n$  osoba koje imaju ime, prezime (stringovi od najviše 50 znakova), spol (znak), broj godina, te osobni identifikacijski broj (OIB). Osobe mogu biti u restoranu s određenim brojem svojih prijatelja.

- (a) Konstruirajte strukturu `osoba` koja omogućava spremanje informacija o određenoj osobi i njezinim prijateljima. Napišite funkciju prototipa `void brisi(osoba* restoran, int n)` koja, iz polja `restoran` i iz svih struktura u njemu, briše sve one osobe koje imaju **barem jednog prijatelja starijeg od 70 godina**.

Rješenje u kojem je za pronalazak prijatelja određene osobe potrebno ispitati sve osobe iz restorana nosi maksimalno 7 bodova.

- (b) Ispišite sve parove osoba, u obliku `(Ime_prve, Prezime_prve - Ime_druge, Prezime_druge)`, koje **imaju najviše 40% zajedničkih prijatelja**.

Zajednički prijatelji dvije osobe  $o_1$  i  $o_2$  su osobe različite od  $o_1, o_2$ , koje poznaju i osobu  $o_1$  i osobu  $o_2$ .

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

### Zadatak 2. (20 bodova)

Napišite funkciju prototipa `char* zamjena(char** rjecnik, int k)` koja vraća pokazivač na string, izgrađen od praznog stringa na sljedeći način. Varijabla `rjecnik` pokazuje na već postojeći niz u kojem je zapisano  $k$  riječi. Korisnik upisuje riječi (po jednu u redu), sve dok ne upiše znak '!'. Za svaku učitane riječ  $r$ , funkcija `zamjena` pretražuje zadani niz od  $k$  riječi i traži prvu riječ  $w$ , takvu da postoje podriječ  $a$  i  $b$ , za koje vrijedi  $r = ab$  i  $w = ba$ . Ako nađe takvu riječ  $w$ , onda tu riječ treba dodati na kraj stringa koji se izgrađuje, tako da je među dodanim riječima po jedan razmak. U suprotnom, string se ne mijenja. Smijete pretpostaviti da učitane riječi nemaju više od 80 znakova.

Primjer: za riječ  $r = \text{karma}$ , tražena riječ može biti  $w = \text{makar}$  (tada je  $a = \text{kar}$ ,  $b = \text{ma}$ ).

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 3.** ( $2 + 6 + 7 = 15$  bodova) Elementi vezane liste trebaju sadržavati podatke o nekim tvrtkama, i to: naziv (s najviše 50 znakova), te vrijednost tvrtke (realni broj).

- (a) Napišite definiciju tipa podatka za pojedini element liste, na način da bude moguće definirati varijable naredbom `tvrtka item;` i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `tvrtka* preuzimanje(tvrtka* first, char* naziv1, char* naziv2)`, gdje je `first` pokazivač na prvi element liste, a `naziv1` je naziv tvrtke koja preuzima tvrtku `naziv2`. Preuzimanje se treba obaviti na način da se tvrtki `naziv1` pribroji vrijednost tvrtke `naziv2`, te da se obriše element vezane liste u kojem su spremljeni podaci tvrtke `naziv2`. Ukoliko barem jedna tvrtka zadanog imena ne postoji u vezanoj listi, funkcija ne radi nikakvu promjenu u ulaznoj vezanoj listi. Funkcija vraća pokazivač na početak vezane liste.
- (c) Napišite funkciju `tvrtka* predstecaj(tvrtka** first)` koja iz liste izdvaja najgorih 5% tvrtki, na način da se formira nova vezana lista, u kojoj su ti elementi spremljeni (sortirani uzlazno po vrijednosti tvrtke), i vraća se pokazivač na početak nove vezane liste. `first` je pokazivač na početak (stare) vezane liste. Broj najgorih 5% tvrtki odredi se tako da se uzme najmanji cijeli broj tvrtki veći ili jednak od 5% ukupnog broja tvrtki. Ukoliko je ulazna lista prazna, funkcija ne radi nikakvu promjenu na ulaznoj vezanoj listi, a vraća `NULL` pokazivač. Ovdje nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima vezane liste.

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 4.** (20 bodova) Deklarirajte strukturu `glas` koja se sastoji (samo) od sljedećih stavki: telefonski broj glasača (string do 15 znakova), ime i prezime natjecatelja (string do 50 znakova) te ocjena (cijeli broj). Niz ovakvih struktura čuva se u binarnoj datoteci `glasanje.dat` i on predstavlja ocjene koje su gledatelji, putem telefona, dali pojedinim natjecateljima u talent showu.

Napišite program koji, za svakog natjecatelja, računa njegovu prosječnu ocjenu i ispisuje sve natjecatelje i njihove prosječne ocjene u tekstualnu datoteku `natjecatelji.txt`, tako da ih sortira silazno po prosjecima. Ispisuje se jedan natjecatelj po liniji, u obliku `"ime prezime: prosjek"` (bez navodnika), a prosjek je zaokružen na tri decimale.

Smijete pretpostaviti da će biti najviše 200 natjecatelja. Smijete koristiti jednu dodatnu strukturu i jedan dodatni niz.

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Rezultati i uvidi u kolokvije:** Rezultati u utorak, 28.6., navečer na webu, a uvidi u srijedu, 29.6., u 11 sati.

**Upute:** Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje, te službeni podsjetnik. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! **Mobitele ugasite i spremite!** Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Ne zaboravite se **potpisati** na svim papirima! Skice smijete raditi i na drugim papirima koje će vam dati dežurni asistent. U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

**Zadatak 1.** (10 + 10 = 20 bodova)

U igraonici se nalazi  $n$  osoba koje imaju ime, prezime (stringovi od najviše 50 znakova), spol (znak), broj godina, te osobni identifikacijski broj (OIB). Osobe mogu biti u igraonici s određenim brojem svojih prijatelja.

- (a) Konstruirajte strukturu `osoba` koja omogućava spremanje informacija o određenoj osobi i njezinim prijateljima. Napišite funkciju prototipa `void brisi(osoba* igraonica, int n)` koja, iz polja `igraonica` i iz svih struktura u njemu, briše sve one osobe koje imaju (strogo) **manje od pet** prijatelja **mlađih od 30** godina.

Rješenje u kojem je za pronalazak prijatelja određene osobe potrebno ispitati sve osobe iz igraonice nosi maksimalno 7 bodova.

- (b) Ispišite sve parove osoba, u obliku `(Ime_prve, Prezime_prve - Ime_druge, Prezime_druge)`, koje **imaju barem jednog** zajedničkog prijatelja **ženskog spola**.

Zajednički prijatelji dvije osobe  $o_1$  i  $o_2$  su osobe različite od  $o_1, o_2$ , koje poznaju i osobu  $o_1$  i osobu  $o_2$ .

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

### Zadatak 2. (20 bodova)

Napišite funkciju prototipa `char* inverzija(char** niz, int l)` koja vraća pokazivač na string, izgrađen od praznog stringa na sljedeći način. Varijabla `niz` pokazuje na već postojeći niz u kojem je zapisano  $l$  riječi. Korisnik upisuje riječi (po jednu u redu), sve dok ne upiše znak `';`. Za svaku učitane riječ  $x$ , funkcija `inverzija` pretražuje zadani niz od  $l$  riječi i traži prvu riječ  $y$ , takvu da postoje  $a$  i  $b$ , podriječi od  $x$ , za koje vrijedi  $x = ab$  i  $y = ab^t$ , gdje je  $b^t$  invertirana (naopaka) riječ od  $b$ . Ako nađe takvu riječ  $y$ , onda tu riječ treba dodati na kraj stringa koji se izgrađuje, tako da je među dodanim riječima po jedan razmak. U suprotnom, string se ne mijenja. Smijete pretpostaviti da učitane riječi nemaju više od 80 znakova.

Primjer: za riječ  $x = \text{iznad}$ , tražena riječ može biti  $y = \text{izdan}$  (tada je  $a = \text{iz}$ ,  $b = \text{nad}$ ).

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 3.** (2 + 6 + 7 = 15 bodova) Elementi vezane liste trebaju sadržavati podatke o autobusnim kompanijama, i to: naziv kompanije (s najviše 40 znakova), te broj autobusa u posjedu tvrtke (prirodni broj).

- (a) Napišite definiciju tipa podatka za pojedini element liste, na način da bude moguće definirati varijable naredbom `autobusi item;` i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- b) Napišite funkciju `autobusi* kupnja(autobusi* first, char* naziv1, char* naziv2)`, gdje je `first` pokazivač na prvi element liste, a `naziv1` je naziv kompanije koja kupuje kompaniju `naziv2`. Kupnja se treba obaviti na način da se kompaniji `naziv1` pribroji broj autobusa kompanije `naziv2`, te da se obriše element vezane liste u kojem su spremljeni podaci kompanije `naziv2`. Ukoliko barem jedna kompanija zadanog imena ne postoji u vezanoj listi, funkcija ne radi nikakvu promjenu u ulaznoj vezanoj listi. Funkcija vraća pokazivač na početak vezane liste.
- (c) Napišite funkciju `autobusi* najbrojniji(autobusi** first)` koja iz liste izdvaja 5% tvrtki s najviše autobusa, na način da se formira nova vezana lista, u kojoj su ti elementi spremljeni (sortirani uzlazno po broju autobusa kompanije), i vraća se pokazivač na početak nove vezane liste. `first` je pokazivač na početak (stare) vezane liste. Broj 5% kompanija s najviše autobusa odredi se tako da se uzme najmanji cijeli broj kompanija veći ili jednak od 5% ukupnog broja kompanija. Ukoliko je ulazna lista prazna, funkcija ne radi nikakvu promjenu na ulaznoj vezanoj listi, a vraća `NULL` pokazivač. Ovdje nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima vezane liste.

## Programiranje 2 – drugi kolokvij, 23. 6. 2016.

**Zadatak 4.** (20 bodova) Deklarirajte strukturu **anketa** koja se sastoji (samo) od sljedećih stavki: šifra ispitanika (string do 10 znakova), naziv proizvoda (string do 100 znakova) te ocjena (cijeli broj). Niz ovakvih struktura čuva se u binarnoj datoteci **ankete.dat** i on predstavlja ocjene koje su ispitanici dali pojedinim proizvodima.

Napišite program koji, za svaki proizvod, računa njegov prosjek ocjena i ispisuje sve proizvode i njihove prosjeke u tekstualnu datoteku **proizvodi.txt**, tako da ih sortira silazno po prosjecima. Ispisuje se jedan proizvod po liniji, u obliku **“naziv: prosjek”** (bez navodnika), a prosjek je zaokružen na tri decimale.

Smijete pretpostaviti da će biti najviše 300 proizvoda. Smijete koristiti jednu dodatnu strukturu i jedan dodatni niz.