

Programiranje 2

2. kolokvij, 2.6.2010.

Rezultati i uvid u kolokvije: ponedjeljak, 7.6.2010, u 12:15; rezultati možda budu dostupni ranije na službenoj stranici kolegija.

Napomene: Sva rješenja napišite isključivo na papir sa zadacima jer jedino njega predajete. Skice smijete raditi na drugim papirima koje će vam dati dežurni asistent.

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje, te službenog podsjetnika. Kalkulatori, razne tablice, papiri, podsjetnici, mobiteli i sl. **nisu dozvoljeni!**

Zadatak 1 (3+10+15+7 bodova) Elementi vezane liste trebaju sadržavati po jedan string `word` od najviše 51 znaka i jedan cijeli broj `num`.

- Napišite definiciju tipa podataka za pojedini element liste, na način da bude moguće definirati varijable naredbom `element item`; i sl. Definirajte samo polja koja su nužna za čuvanje takve liste u memoriji.
- Napišite funkciju `void output(element *first)` (`first` je pointer na prvi element liste) koja će ispisati aritmetičku sredinu svih `num`-ova u onim elementima liste u kojima pripadna riječ `word` nije palindrom (različito se čita slijeva nadesno i zdesna nalijevo).
- Napišite funkciju `del()` koja iz liste briše sve elemente kojima je riječ jednaka nekoj zadanoj riječi (zadaje se kao argument funkcije), te na neki način vraća i broj obrisanih elemenata i sumu njihovih `num`-ova. Deklaraciju funkcije `del()` (dakle, argumente i povratni tip) sami odredite, te navedite primjer poziva funkcije uz kraća objašnjenja što je koji parametar.
- Napišite funkciju `element *input(const char name[])`; koja će iz tekstualne datoteke `name` čitati podatke, te od njih kreirati novu listu na način da im redoslijed bude suprotan od onoga u datoteci. U datoteci su redom po jedna riječ (`word` pojedinog elementa) i jedan cijeli broj (`num` pojedinog elementa), odvojeni razmacima. Možete pretpostaviti da niti jedan tekstualni dio u datoteci nije predugačak. Funkcija treba vratiti pointer na prvi element liste.

Napomene:

- Prilikom svih alokacija memorije i otvaranja datoteka povjerite da su te operacije zaista i uspjele. Ako ne uspiju, prekinite izvršavanje funkcije (ali ne i programa).
- Podzadatke b)–d) ne morate rješavati redom kojim su zadani, ali je nužno da prvo riješite podzadatak a).

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 2 (1+3+1+5 bodova) U bistrou “Žabac” ponestalo je pića na skladištu. Vlasnik bistroa sastavio je popis narudžbi, te je dobio ponude od svojih dobavljača.

- Definirajte strukturu `Cijena`, koja se sastoji od dva broja tipa `unsigned`, koji predstavljaju iznos u kunama i lipama, respektivno.
- Napišite funkciju s prototipom:

```
int usp_cijene(const Cijena *c1, const Cijena *c2);
```

Funkcija treba vratiti -1 ako je cijena `*c1` strogo manja od cijene `*c2`, 0 ako su cijene `*c1` i `*c2` jednake, odnosno 1 ako je cijena `*c1` strogo veća od cijene `*c2`.

- Definirajte strukturu `Ponuda` koja se sastoji od OIBa dobavljača (11 znakova + terminator), naziva (31 znak + terminator) pića, te cijene (tipa `Cijena`) za bocu pića.
- Napišite funkciju s prototipom:

```
int usp_ponude(const Ponuda *p1, const Ponuda *p2);
```

Funkcija treba vratiti -1 ako:

- naziv pića u ponudi `*p1` je leksikografski strogo prije naziva pića u ponudi `*p2`, ili
- nazivi pića su jednaki, ali je jedinična cijena u ponudi `*p1` strogo manja od cijene u ponudi `*p2`.

Inače, ako su nazivi i cijene pića jednaki, funkcija treba vratiti 0 , u suprotnom treba vratiti 1 .

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 3 (13+17 bodova) Koristeći strukture i funkcije iz prethodnog zadatka, napišite:

- a) [7(+6) bodova] funkciju s prototipom:

```
int nadji(FILE *f, const char *naziv);
```

koja u datoteci `*f` sa zapisima tipa `Ponuda`, otvorenoj za binarno čitanje i **sortiranoj** prema uređaju definiranom funkcijom `usp_ponude`, pronalazi prvi zapis s danim nazivom pića, **bez** korištenja pomoćnih polja. Upotrijebite li algoritam optimalne složenosti, dobivate dodatnih 6 bodova. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije pokazivač se treba nalaziti na početku pronađenog zapisa, ili na kraju datoteke ako odgovarajući zapis nije pronađen. Funkcija treba vratiti redni broj, počev od 0, pronađenog zapisa, ili EOF ako zapis nije pronađen.

- b) [12 ili 17 bodova] funkciju s prototipom:

```
int sortiraj(FILE *f);
```

koja sortira datoteku `*f` otvorenu za binarno pisanje i čitanje. Datoteka se sastoji od zapisa tipa `Ponuda`. Za usporedbu zapisa koristite uređaj definiran funkcijom `usp_ponude`. Funkcija treba vratiti broj zapisa u datoteci. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije treba biti 0 (početak datoteke). Sortiranje možete provesti izravno na datoteci (17 bodova) ili pomoću dinamički alociranog polja (12 bodova).

Programiranje 2

2. kolokvij, 2.6.2010.

Rezultati i uvid u kolokvije: ponedjeljak, 7.6.2010, u 12:15; rezultati možda budu dostupni ranije na službenoj stranici kolegija.

Napomene: Sva rješenja napišite isključivo na papir sa zadacima jer jedino njega predajete. Skice smijete raditi na drugim papirima koje će vam dati dežurni asistent.

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje, te službenog podsjetnika. Kalkulatori, razne tablice, papiri, podsjetnici, mobiteli i sl. **nisu dozvoljeni!**

Zadatak 1 (3+15+10+7 bodova) Elementi vezane liste trebaju sadržavati po dva stringa `num` i `word` od po najviše 73 znaka.

- Napišite definiciju tipa podataka za pojedini element liste. Pri tome, definirajte samo polja koja su nužna za čuvanje takve liste u memoriji, te pazite da bude moguće definirati varijable naredbom `element elt;`.
- Napišite funkciju `remove()` koja iz liste briše prvi element kojem je string `word` leksikografski nakon ("veći" od) stringa `num`. Funkcija treba na neki način vratiti i broj obrisanih elemenata (0 ili 1) i ukupni broj slova (u oba stringa) za izbrisani element (ili nulu ako ništa nije izbrisano). Deklaraciju funkcije `remove()` (dakle, argumente i povratni tip) sami odredite, te objasnite kako se funkcija poziva (primjer i kraće objašnjenje što je koji parametar).
- Napišite funkciju `void writeout(element *first)` koja će ispisati produkt duljina polja `word` svih onih elemenata liste (zadane preko pointera `first` na prvi element) u kojima je pripadni string `num` palindrom (jednako se čita slijeva nadesno i zdesna nalijevo).
- Napišite funkciju `element *read(const char name[]);` koja će iz tekstualne datoteke `name` čitati podatke, te od njih kreirati novu listu. Pri tome, svaki element u datoteci zauzima po dva retka, pri čemu se prvi redak učitava kao `num`, a drugi kao `word`. Možete pretpostaviti da niti jedan tekstualni dio u datoteci nije predugačak. Funkcija treba vratiti pointer na prvi element liste u kojoj će redosljed elemenata odgovarati redosljedu tih podataka u datoteci.

Napomene:

- Prilikom svih alokacija memorije i otvaranja datoteka povjerite da su te operacije zaista i uspjele. Ako ne uspiju, prekinite izvršavanje funkcije (ali ne i programa).
- Podzadatke b)–d) ne morate rješavati redom kojim su zadani, ali je nužno da prvo riješite podzadatak a).

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 2 (2+3+5 bodova) Kružnice u koordinatnoj ravnini.

a) Definirajte strukture: **Točka**, koja se sastoji od x i y koordinata tipa `double`, te **Kružnica**, koja se sastoji od jedne **Točke** (središta) i radijusa tipa `double`.

b) Napišite funkciju s prototipom:

```
double udalj_toc(const Točka *t1, const Točka *t2);
```

Funkcija treba vratiti euklidsku udaljenost točaka `*t1` i `*t2`.

c) Napišite funkciju s prototipom:

```
int usp_kr(const Kružnica *k1, const Kružnica *k2);
```

Funkcija treba vratiti `-1` ako:

- središte kružnice `*k1` je strogo bliže ishodištu od središta kružnice `*k2`, ili
- radijus kružnice `*k1` je strogo manji od radijusa kružnice `*k2`.

Inače, ako su središta kružnica jednako udaljena od ishodišta i radijusi su jednaki, funkcija treba vratiti `0`, u suprotnom treba vratiti `1`.

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 3 (13+17 bodova) Koristeći strukture i funkcije iz prethodnog zadatka, napišite:

- a) [7(+6) bodova] funkciju s prototipom:

```
int nadji(FILE *f, const Tocka *sred);
```

koja u datoteci `*f` sa zapisima tipa `Kruzica`, otvorenoj za binarno čitanje i **sortiranoj** prema uređaju definiranom funkcijom `usp_kr`, pronalazi prvi zapis s danim središtem kružnice, **bez** korištenja pomoćnih polja. Upotrijebite li algoritam optimalne složenosti, dobivate dodatnih 6 bodova. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije pokazivač se treba nalaziti na početku pronađenog zapisa, ili na kraju datoteke ako odgovarajući zapis nije pronađen. Funkcija treba vratiti redni broj, počev od 0, pronađenog zapisa, ili EOF ako zapis nije pronađen.

- b) [12 ili 17 bodova] funkciju s prototipom:

```
int sortiraj(FILE *f);
```

koja sortira datoteku `*f` otvorenu za binarno pisanje i čitanje. Datoteka se sastoji od zapisa tipa `Kruzica`. Za usporedbu zapisa koristite uređaj definiran funkcijom `usp_kr`. Funkcija treba vratiti broj zapisa u datoteci. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije treba biti 0 (početak datoteke). Sortiranje možete provesti izravno na datoteci (17 bodova) ili pomoću dinamički alociranog polja (12 bodova).

Programiranje 2

2. kolokvij, 2.6.2010.

Rezultati i uvid u kolokvije: ponedjeljak, 7.6.2010, u 12:15; rezultati možda budu dostupni ranije na službenoj stranici kolegija.

Napomene: Sva rješenja napišite isključivo na papir sa zadacima jer jedino njega predajete. Skice smijete raditi na drugim papirima koje će vam dati dežurni asistent.

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje, te službenog podsjetnika. Kalkulatori, razne tablice, papiri, podsjetnici, mobiteli i sl. **nisu dozvoljeni!**

Zadatak 1 (3+15+7+10 bodova) Elementi vezane liste trebaju sadržavati po dva stringa `rijec` i `br` od po najviše 83 znaka.

- Napišite definiciju tipa podataka za pojedini element liste. Pri tome, definirajte samo polja koja su nužna za čuvanje takve liste u memoriji, te pazite da bude moguće definirati varijable naredbom `element nesto;`.
- Napišite funkciju `brisi()` koja iz liste briše sve element kojima je string `rijec` leksikografski prije ("manji" od) stringa `br`. Funkcija treba na neki način vratiti i broj obrisanih elemenata i ukupni broj znamenaka (u oba stringa) u svim izbrisanim elementima. Deklaraciju funkcije `brisi()` (dakle, argumente i povratni tip) sami odredite, te objasnite kako se funkcija poziva (primjer i kraće objašnjenje što je koji parametar).
- Napišite funkciju `element *citanje(const char ime[])`; koja će iz tekstualne datoteke `ime` čitati podatke, te od njih kreirati novu listu na način da im redosljed bude suprotan od onoga u datoteci. Svaki element u datoteci zauzima svoj redak, pri čemu se prva riječ učitava kao `rijec`, a ostatak linije kao `br`. Možete pretpostaviti da niti jedan tekstualni dio u datoteci nije predugačak. Funkcija treba vratiti pointer na prvi element liste.
- Napišite funkciju `void ispis(element *prvi)` koja će ispisati sumu duljina polja `rijec` svih onih elemenata liste (zadane preko pointera `prvi` na prvi element) u kojima pripadni string `br` nije palindrom (različito se čita slijeva nadesno i zdesna nalijevo).

Napomene:

- Prilikom svih alokacija memorije i otvaranja datoteka povjerite da su te operacije zaista i uspjele. Ako ne uspiju, prekinite izvršavanje funkcije (ali ne i programa).
- Podzadatke b)–d) ne morate rješavati redom kojim su zadani, ali je nužno da prvo riješite podzadatak a).

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 2 (1+3+1+5 bodova) Uplate u mirovinski fond.

- Definirajte strukturu `Uplata`, koja se sastoji od dva elementa tipa `unsigned`, koji predstavljaju iznos u kunama i lipama, respektivno.
- Napišite funkciju s prototipom:

```
int usp_uplate(const Uplata *u1, const Uplata *u2);
```

Funkcija treba vratiti -1 ako je iznos uplate `*u1` strogo manji od iznosa uplate `*u2`, 0 ako su iznosi uplata `*u1` i `*u2` jednaki, odnosno 1 ako je iznos uplate `*u1` strogo veći od iznosa uplate `*u2`.

- Definirajte strukturu `Nalog` koja se sastoji od OIBa mirovinskog fonda (11 znakova + terminator), šifre uplatitelja (tipa `unsigned`), te iznosa uplate (tipa `Uplata`).
- Napišite funkciju s prototipom:

```
int usp_naloge(const Nalog *n1, const Nalog *n2);
```

Funkcija treba vratiti -1 ako:

- iznos uplate u nalogu `*n1` je strogo manji od iznosa uplate u nalogu `*n2`, ili
- iznosi uplata su jednaki, ali je OIB u nalogu `*n1` leksikografski strogo prije OIBa u nalogu `*n2`.

Inače, ako su iznosi uplata i OIBi jednaki, funkcija treba vratiti 0 , u suprotnom treba vratiti 1 .

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 3 (13+17 bodova) Koristeći strukture i funkcije iz prethodnog zadatka, napišite:

- a) [7(+6) bodova] funkciju s prototipom:

```
int nadji(FILE *f, const Uplata *u);
```

koja u datoteci `*f` sa zapisima tipa `Nalog`, otvorenoj za binarno čitanje i **sortiranoj** prema uređaju definiranom funkcijom `usp_naloge`, pronalazi prvi zapis s danim iznosom uplate, **bez** korištenja pomoćnih polja. Upotrijebite li algoritam optimalne složenosti, dobivate dodatnih 6 bodova. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije pokazivač se treba nalaziti na početku pronađenog zapisa, ili na kraju datoteke ako odgovarajući zapis nije pronađen. Funkcija treba vratiti redni broj, počev od 0, pronađenog zapisa, ili EOF ako zapis nije pronađen.

- b) [12 ili 17 bodova] funkciju s prototipom:

```
int sortiraj(FILE *f);
```

koja sortira datoteku `*f` otvorenu za binarno pisanje i čitanje. Datoteka se sastoji od zapisa tipa `Nalog`. Za usporedbu zapisa koristite uređaj definiran funkcijom `usp_naloge`. Funkcija treba vratiti broj zapisa u datoteci. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije treba biti 0 (početak datoteke). Sortiranje možete provesti izravno na datoteci (17 bodova) ili pomoću dinamički alociranog polja (12 bodova).

Programiranje 2

2. kolokvij, 2.6.2010.

Rezultati i uvid u kolokvije: ponedjeljak, 7.6.2010, u 12:15; rezultati možda budu dostupni ranije na službenoj stranici kolegija.

Napomene: Sva rješenja napišite isključivo na papir sa zadacima jer jedino njega predajete. Skice smijete raditi na drugim papirima koje će vam dati dežurni asistent.

Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje, te službenog podsjetnika. Kalkulatori, razne tablice, papiri, podsjetnici, mobiteli i sl. **nisu dozvoljeni!**

Zadatak 1 (3+10+7+15 bodova) Elementi vezane liste trebaju sadržavati po jedan prirodni broj **br** i string **rijec** od najviše 71 znaka.

- Napišite definiciju tipa podataka za pojedini element liste. Pri tome, definirajte samo polja koja su nužna za čuvanje takve liste u memoriji, te pazite da bude moguće definirati varijable naredbom `element nesto;`.
- Napišite funkciju `void ispis(element *prvi)` (`prvi` je pointer na prvi element liste) koja će ispisati geometrijsku sredinu svih `br`-ova u onim elementima liste u kojima je pripadna riječ `rijec` palindrom (jednako se čita slijeva nadesno i zdesna nalijevo).
- Napišite funkciju `element *citanje(const char ime[])`; koja će iz tekstualne datoteke `ime` čitati podatke, te od njih kreirati novu listu na način da im redosljed odgovara onome u datoteci. U datoteci se nalaze redom po jedan broj (polje `br` pojedinog elementa) i jedna riječ (polje `rijec` pojedinog elementa). Ti elementi odvojeni su razmacima. Možete pretpostaviti da riječi u datoteci nisu predugačke. Funkcija treba vratiti pointer na prvi element liste.
- Napišite funkciju `brisi()` koja iz liste briše prvi element kojem je riječ `rijec` različita od neke zadane riječi (zadaje se kao argument funkcije), te na neki način vraća i broj obrisanih elemenata (nula ili jedan) i njegov `num` (ili `-1` ako ništa nije izbrisano). Deklaraciju funkcije `brisi()` (dakle, argumente i povratni tip) sami odredite, te objasnite kako se funkcija poziva (primjer i kraće objašnjenje što je koji parametar).

Napomene:

- Prilikom svih alokacija memorije i otvaranja datoteka povjerite da su te operacije zaista i uspjele. Ako ne uspiju, prekinite izvršavanje funkcije (ali ne i programa).
- Podzadatke b)–d) ne morate rješavati redom kojim su zadani, ali je nužno da prvo riješite podzadatak a).

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 2 (1+3+1+5 bodova) Novčići kao valjci.

a) Definirajte strukturu `Valjak`, koja se sastoji od dva elementa tipa `double`, koji predstavljaju radijus baze i visinu valjka.

b) Napišite funkciju s prototipom:

```
double volumen(const Valjak *v);
```

Funkcija treba vratiti volumen valjka `*v` ili 0.0 ako su komponente valjka pogrešno zadane.

c) Definirajte strukturu `Novcic`, koja se sastoji od `Valjka` i mase novčića tipa `double`.

d) Napišite funkciju s prototipom:

```
int usp_nov(const Novcic *n1, const Novcic *n2);
```

Funkcija treba vratiti `-1` ako:

- volumen novčića `*n1` je strogo manji od volumena novčića `*n2`, ili
- volumeni novčića su isti, ali masa novčića `*n1` je strogo manja od mase novčića `*n2`.

Inače, ako su volumeni i mase novčića jednaki, funkcija treba vratiti 0, u suprotnom treba vratiti 1.

Programiranje 2

2. kolokvij, 2.6.2010.

Zadatak 3 (13+17 bodova) Koristeći strukture i funkcije iz prethodnog zadatka, napišite:

- a) [7(+6) bodova] funkciju s prototipom:

```
int najdi(FILE *f, double v);
```

koja u datoteci ***f** sa zapisima tipa **Novcic**, otvorenoj za binarno čitanje i **sortiranoj** prema uređaju definiranom funkcijom **usp_nov**, pronalazi prvi zapis s danim volumenom novčića, **bez** korištenja pomoćnih polja. Upotrijebite li algoritam optimalne složenosti, dobivate dodatnih 6 bodova. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije pokazivač se treba nalaziti na početku pronađenog zapisa, ili na kraju datoteke ako odgovarajući zapis nije pronađen. Funkcija treba vratiti redni broj, počev od 0, pronađenog zapisa, ili EOF ako zapis nije pronađen.

- b) [12 ili 17 bodova] funkciju s prototipom:

```
int sortiraj(FILE *f);
```

koja sortira datoteku ***f** otvorenu za binarno pisanje i čitanje. Datoteka se sastoji od zapisa tipa **Novcic**. Za usporedbu zapisa koristite uređaj definiran funkcijom **usp_nov**. Funkcija treba vratiti broj zapisa u datoteci. Vrijednost datotečnog pokazivača na početku je nedefinirana, a na kraju rada funkcije treba biti 0 (početak datoteke). Sortiranje možete provesti izravno na datoteci (17 bodova) ili pomoću dinamički alociranog polja (12 bodova).