JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Rezultati i uvidi: Rezultati u srijedu, 9. 9. 2020., navečer na webu, a uvidi u četvrtak, 10. 9. 2020., u 10 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 isključite i spremite**! Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Obavezno predajte **sve** papire sa zadacima, čak i ako neke zadatke niste rješavali. 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. Dozvoljeno je pisanje pomoćnih funkcija.

Zadatak 1. (30 bodova) Za broj $m \in \mathbb{N}$ kažemo da je *uravnotežen* ako je apsolutna vrijednost razlike eksponenta najveće potencije od 2 koja dijeli m i eksponenta najveće potencije od 3 koja dijeli m jednaka 1. Napišite rekurzivnu funkciju koja prima nenegativan prirodan broj n (i možda još neke parametre) i vraća broj načina na koji se n može napisati kao suma uravnoteženih brojeva. Raspise koji se razlikuju do na poredak pribrojnika smatramo istima. Obavezno napišite i poziv funkcije.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Zadatak 2. (25 bodova) Razni viši programski jezici podržavaju *lisne rezove*, odnosno dozvoljavaju vam za listu L napisati $L[2:5]$ (ili nešto slično) kako biste dobili listu koja sadrži sve elemente liste L od trećeg do šestog elementa (za potrebe ovog zadatka, pretpostavljamo da je prvi element liste na poziciji 0 te da su elementi na oba ruba uključeni; kod nekih jezika su te pretpostavke drugačije).

(a) Napišite deklaraciju tipa `lista` za vezanu listu cijelih brojeva. Zatim napišite funkciju `lista rez(lista L, int l, int d)` koja prima vezanu listu i dva cijela broja koji predstavljaju željene rubne pozicije te vraća lisni rez, odnosno novu vezanu listu; ona sadrži sve elemente liste `L` od elementa na poziciji `l` do elementa na poziciji `d` (preciznije, ali i dulje bi bilo pisati — njihove kopije). Ako je `L` neprazna, možete pretpostaviti da su rubovi `l` i `d` u skupu $\{0, \ldots, \text{duljina}(\texttt{L}) - 1\}$. Rez je, dakako, prazna lista ako je `d` $<$ `l`. Primjerice, za listu `L` duljine barem 4, poziv `rez(L,1,3)` vraća listu duljine 3 koja redom sadrži drugi, treći i četvrti element liste `L` (elemente na pozicijama 1,2,3).

(b) Napišite funkciju koja prima listu L cijelih brojeva i prirodan broj n te omogućuje korisniku unijeti cijele brojeve $l_1, d_1, \ldots, l_n, d_n$. Zatim pomoću funkcije iz (a) ispisuje **bilo kojim redoslijedom** sve brojeve iz „unije" lisnih rezova $L[l_1 : d_1], \ldots, L[l_n : d_n]$ (dakle, sve brojeve koji se pojavljuju u barem jednom od rezova, i samo njih, ispiše točno jednom). Ako je unija prazna, treba ispisati odgovarajuću poruku. Primjerice, za listu $1, 2, 1, 3$, n jednak 3 i rubove $0, 3, 0, 2, 1, 1$ treba ispisati $1, 2, 3$ u proizvoljnom redoslijedu. Kao i pod (a), nije potrebno provjeravati ispravnost argumenata funkcije. Dozvoljeno je koristiti **jednu** dodatnu listu.

Napomena: U oba podzadatka, početne liste moraju ostati nepromijenjene.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Zadatak 3. (30 bodova) Dan je rječnik p realiziran kao polje pokazivača na stringove pri čemu je posljednji string u rječniku prazan string, tj. posljednji pokazivač pokazuje na `'\0'`. Rječnik je dinamički alociran.

(a) Napišite funkciju `int provjeriSort(char **p)` koja provjerava je li rječnik sortiran uzlazno. Ako jest, vraća broj riječi u rječniku, a inače vraća vrijednost -1.

(b) Napišite funkciju `void dodajString(char **p,char *str)` koja prima uzlazno sortiran rječnik `p` i string `str`. Ako string `str` nije u rječniku, funkcija ga treba ubaciti u rječnik na odgovarajuće mjesto tako da rječnik ostane sortiran. Ako se string već nalazi u rječniku, tada funkcija ne radi ništa. Možete pretpostaviti da ima dovoljno alocirane memorije za dodavanje stringa u rječnik.

(c) Napišite funkciju `char *izmijeniString(char *str, char *staro, char *novo)` koja prima tri stringa. Funkcija treba stvoriti novi string iz stringa `str` pri čemu svaku pojavu stringa `staro` (kao podstring, i to slijeva nadesno) treba zamijeniti stringom `novo`. Npr. za `str="Moja je zgrada na kraju sela"`, `staro="zgrada"`, te `novo="kuca"` funkcija treba vratiti string `"Moja je kuca na kraju sela"`.

Smijete koristiti zaglavlje `string.h`.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Zadatak 4. (30 bodova) U trenutnom su direktoriju spremljene tekstualne datoteke s imenima oblika `c.txt`, gdje je `c` veliko slovo engleske abecede (`A.txt`, `B.txt`, ..., `Z.txt`). Neke od tih datoteka sadrže specijalne `#include` naredbe koje su uvijek oblika `#include <c.txt>`.

Napišite funkciju `void posao(char c, char *adresa)` koja će u datoteku na adresi `adresa` (argument funkcije) ispisati sadržaj datoteke `c.txt` (`c` je znak koji je argument funkcije). Pritom na mjestima gdje se u datoteci `c.txt` javlja niz znakova oblika `#include <IME>` treba ispisati sadržaj datoteke `IME`. Pretpostavite da se u datoteci `IME` ne javlja naredba `#include`. Datoteke su takve da se ne može dogoditi da opisani proces rezultira pojavom naredbi koje nisu bile prisutne u izvornoj datoteci `c.txt`.

Primjer ulaznih datoteka i izlazne datoteke nakon poziva `posao('D', "izlaz.txt")`:

B.txt:
```
Ja sam iz B.txt
I ja sam iz B.txt
```

D.txt:
```
Ja sam iz D.txt
#include <B.txt>
I ja sam iz D.txt
123#include <G.txt>456
#inclu <H.txt>
```

G.txt:
```
Ja sam iz G.txt
```

izlaz.txt:
```
Ja sam iz D.txt
Ja sam iz B.txt
I ja sam iz B.txt
I ja sam iz D.txt
123Ja sam iz G.txt456
#inclu <H.txt>
```

Napomene:

- sve `#include` naredbe su sintaksno ispravno zapisane, tj. nakon niza znakova `#include` uvijek slijedi jedan razmak i potom niz oblika `<c.txt>`
- možete koristiti pomoćne nizove;
- ne trebate provjeravati uspješnost funkcija za rad s datotekama;
- ulazne datoteke možete otvarati i zatvarati proizvoljan broj puta.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Rezultati i uvidi: Rezultati u srijedu, 9. 9. 2020., navečer na webu, a uvidi u četvrtak, 10. 9. 2020., u 10 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 isključite i spremite**! Sva rješenja napišite isključivo na papire sa zadacima, jer jedino njih predajete. Obavezno predajte **sve** papire sa zadacima, čak i ako neke zadatke niste rješavali. 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. Dozvoljeno je pisanje pomoćnih funkcija.

Zadatak 1. (30 bodova) Za broj $m \in \mathbb{N}$ kažemo da je *uravnotežen* ako je apsolutna vrijednost razlike eksponenta najveće potencije od 5 koja dijeli m i eksponenta najveće potencije od 7 koja dijeli m jednaka 1. Napišite rekurzivnu funkciju koja prima nenegativan prirodan broj n (i možda još neke parametre) i vraća broj načina na koji se n može napisati kao suma uravnoteženih brojeva. Raspise koji se razlikuju do na poredak pribrojnika smatramo istima. Obavezno napišite i poziv funkcije.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Zadatak 2. (25 bodova) Razni viši programski jezici podržavaju *lisne rezove*, odnosno dozvoljavaju vam za listu L napisati $L[1:8]$ (ili nešto slično) kako biste dobili listu koja sadrži sve elemente liste L od drugog do devetog elementa (za potrebe ovog zadatka, pretpostavljamo da je prvi element liste na poziciji 0 te da su elementi na oba ruba uključeni; kod nekih jezika su te pretpostavke drugačije).

(a) Napišite deklaraciju tipa `lista` za vezanu listu realnih brojeva. Zatim napišite funkciju `lista rez(lista L, int l, int d)` koja prima vezanu listu i dva cijela broja koji predstavljaju željene rubne pozicije te vraća lisni rez, odnosno novu vezanu listu; ona sadrži sve elemente liste `L` od elementa na poziciji `l` do elementa na poziciji `d` (preciznije, ali i dulje bi bilo pisati — njihove kopije). Ako je `L` neprazna, možete pretpostaviti da su rubovi `l` i `d` u skupu $\{0, \ldots, \text{duljina}(\texttt{L}) - 1\}$. Rez je, dakako, prazna lista ako je `d` $<$ `l`. Primjerice, za listu `L` duljine barem 5, poziv `rez(L,1,4)` vraća listu duljine 4 koja redom sadrži drugi, treći, četvrti i peti element liste `L` (elemente na pozicijama $1, 2, 3, 4$).

(b) Napišite funkciju koja prima listu L realnih brojeva i prirodan broj n kao argumente te omogućuje korisniku unijeti cijele brojeve $l_1, d_1, \ldots, l_n, d_n$. Zatim pomoću funkcije iz (a) ispisuje **bilo kojim redoslijedom** sve brojeve iz „presjeka" rezova $L[l_1 : d_1], \ldots, L[l_n : d_n]$ (dakle, sve brojeve koji se pojavljuju u svim navedenim rezovima, i samo njih, ispiše točno jednom). Ako je presjek prazan, treba ispisati odgovarajuću poruku. Primjerice, za listu $5.2, 6, 6, 5.2, 8$, n jednak 2 i rubove $1, 4, 0, 2$ treba ispisati $5.2, 6$ u proizvoljnom redoslijedu. Kao i pod (a), nije potrebno provjeravati ispravnost argumenata funkcije. Dozvoljeno je koristiti **jednu** dodatnu listu.

Napomena: U oba podzadatka, početne liste moraju ostati nepromijenjene.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Zadatak 3. (30 bodova) Dan je rječnik p realiziran kao polje pokazivača na stringove pri čemu je posljednji string u rječniku prazan string, tj. posljednji pokazivač pokazuje na `'\0'`. Rječnik je dinamički alociran.

(a) Napišite funkciju `int provjeriSort(char **p)` koja provjerava je li rječnik sortiran silazno. Ako jest, vraća broj riječi u rječniku, a inače vraća vrijednost -1.

(b) Napišite funkciju `void izbaciString(char **p,char *str)` koja prima silazno sortiran rječnik `p` i string `str` kojeg treba izbaciti iz rječnika ako se nalazi u njemu. Ne trebate oslobađati memoriju nakon izbacivanja.

(c) Napišite funkciju `char *izmijeniString(char *str, char *staro, char *novo)` koja prima tri stringa. Funkcija treba stvoriti novi string iz stringa `str` pri čemu svaku pojavu stringa `staro` (kao podstring, i to slijeva nadesno) treba zamijeniti stringom `novo`. Npr. za `str="Moja je zgrada na kraju sela"`, `staro="zgrada"`, te `novo="kuca"` funkcija treba vratiti string `"Moja je kuca na kraju sela"`.

Smijete koristiti zaglavlje `string.h`.

JMBAG

IME I PREZIME

Programiranje 2 – prva popravna provjera znanja, 4. 9. 2020.

Zadatak 4. (30 bodova) U trenutnom su direktoriju spremljene tekstualne datoteke s imenima oblika `n.txt`, gdje je `n` dekadska znamenka (`0.txt`, `1.txt`, ..., `9.txt`). Neke od tih datoteka sadrže specijalne `#include` naredbe koje su uvijek oblika `#include <n.txt>`.

Napišite funkciju `void ispis(int n)` koja će u datoteku na adresi `sve.txt` ispisati sadržaj datoteke `n.txt` (`n` je argument funkcije). Pritom na mjestima gdje se u datoteci `n.txt` javlja niz znakova oblika `#include <IME>` treba ispisati sadržaj datoteke `IME`. Pretpostavite da se u datoteci `IME` ne javlja naredba `#include`. Datoteke su takve da se ne može dogoditi da opisani proces rezultira pojavom naredbi koje nisu bile prisutne u izvornoj datoteci `n.txt`.

Primjer ulaznih datoteka i izlazne datoteke nakon poziva `ispis(3)`:

2.txt:
```
Ja sam iz 2.txt
I ja sam iz 2.txt
```

3.txt:
```
Ja sam iz 3.txt
ABC#include <7.txt>DEF
I ja sam iz 3.txt
#include <2.txt>
#inclu <9.txt>
```

7.txt:
```
Ja sam iz 7.txt
```

izlaz.txt:
```
Ja sam iz 3.txt
ABCJa sam iz 7.txtDEF
I ja sam iz 3.txt
Ja sam iz 2.txt
I ja sam iz 2.txt
#inclu <9.txt>
```

Napomene:

- sve `#include` naredbe su sintaksno ispravno zapisane, tj. nakon niza znakova `#include` uvijek slijedi jedan razmak i potom niz oblika `<n.txt>`
- možete koristiti pomoćne nizove;
- ne trebate provjeravati uspješnost funkcija za rad s datotekama;
- ulazne datoteke možete otvarati i zatvarati proizvoljan broj puta.