

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Rezultati i uvidi u kolokvije:** Rezultati u četvrtak, 28.6., navečer na webu, a uvidi u petak, 29.6., u 13 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.** (12 + 8 bodova) Svakom stringu  $s$  pridružujemo vrijednost (tipa `double`) funkcijom  $f(s) = \sum_i i \cdot \text{ascii}(s[i])$ , gdje `ascii` označava ASCII kôd znaka  $s[i]$ , a suma ide po svim indeksima  $i$  za koje je  $s[i] \neq '\0'$ . Niz stringova zadan je poljem pokazivača na stringove (prve znakove).

- Napišite funkciju `int evaluiraj_i_promijeni(char ** polje, int n)` koja, za ulazni niz stringova `polje` i duljinu polja `n`, računa prosječnu vrijednost funkcije  $f$  na svim stringovima iz tog polja. Za sve stringove  $s$ , za koje je  $f(s)$  strogo **manje** od prosjeka, treba izbaciti pripadni pokazivač iz polja `polje` (`polje` ne treba realocirati). Tako promijenjeno polje treba sortirati **uzlazno** prema vrijednosti funkcije  $f$ , koristeći samo zamjene pokazivača. Funkcija vraća novu duljinu polja `polje`.
- Napišite funkciju `kreiraj` koja prima jedno polje znakova, u kojem je spremljen neki (nepoznati) broj stringova, tako da su stringovi nanizani jedan za drugim (kao niz znakova), a na kraju niza nalazi se string "`kraj`". Funkcija treba kreirati (alocirati) pripadno polje **pokazivača** na prvi znak svakog stringa u nizu (bez završnog stringa "`kraj`"). Funkcija vraća pokazivač na prvi element tog polja (ili `NULL`, ako je polje prazno), a kroz varijabilni argument treba vratiti broj pokazivača u tom polju (tj. broj stringova u ulaznom nizu, bez završnog). Alocirajte točno onliko memorije koliko treba za spremanje polja pokazivača. Dozvoljeno je koristiti `malloc` i `realloc`.

**Napomene:** Zabranjeno je koristiti dodatna polja, osim navedenih. Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 2.** (20 bodova) Definirajte tip podataka `jelo` kao strukturu koja sadrži sljedeće podatke: string `naziv`, polje stringova `namirnice`, te `duljina` polja i `cijena` kao cjelobrojni tip. Duljina svih stringova je ograničena na 100 znakova.

- (a) Napišite funkciju `kreiraj`, koja alocira strukturu za novo jelo, učitava sve navedene podatke sa standardnog ulaza, i vraća pokazivač na alociranu strukturu.
- b) Napišite funkciju `nutricionist`, koja prima polje jela i sa standardnog ulaza učitava podatke “idealnog” jela. Zatim, u polju traži jelo koje ima najviše zajedničkih namirnica s idealnim jelom. Ako takvih jela ima više, onda se uzima najjeftinije jelo. Ako i takvih ima više, uzima se ono najmanjeg indeksa. Na mjesto nađenog jela (u polju) treba upisati idealno jelo, a staro (ono nađeno) treba vratiti kroz varijabilni argument. Funkcija vraća 1 ako staro jelo ima sve namirnice idealnog, a 0 inače.

**Napomene:** Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 3.** (2 + 5 + 10 + 3 bodova) Podatke o restoranima u jednoj ulici spremamo u vezanu listu. Za svaki restoran pamtimo naziv (string od najviše 50 znakova), kućni broj (prirodni broj) i zaradu restorana (realni broj). Vezana lista restorana je uvijek *uzlazno* sortirana po kućnom broju. Možete pretpostaviti da ne postoje dva restorana s istim kućnim brojem.

- (a) Napišite definiciju odgovarajućeg tipa podatka za restoran, na način da bude moguće definirati varijable naredbom `rest a`; i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
  - (b) Napišite funkciju `rest* unesi(rest* first, rest* new)` koja prima vezanu listu restorana (`first` je pokazivač na početak ulazne vezane liste) te restoran `new`. Funkcija ubacuje element `new` u vezanu listu tako da vezana lista i dalje bude *uzlazno* sortirana po kućnom broju. Ulazna vezana lista može biti prazna. Funkcija vraća pokazivač na prvi element dobivene vezane liste, nakon ubacivanja novog elementa.
  - (c) Napišite funkciju `rest* prvih_k(rest** first, int k)` koja traži prvih `k` susjednih restorana s ukupnom najvećom zaradom i izbacuje ih iz vezane liste. Dva restorana su susjedna ako u vezanoj listi ne postoji restoran koji se nalazi između njih. Funkcija vraća pokazivač na početak vezane liste izbačenih restorana (koja, također, mora biti sortirana *uzlazno* po kućnim brojevima). Ukoliko u ulaznoj vezanoj listi ima strogo manje od `k` elemenata, funkcija ne mijenja ulaznu vezanu listu i vraća `NULL`.
  - (d) U glavnom programu formirajte vezanu listu od 5 restorana (elemenata), koristeći funkciju `unesi`. Podatke za te restorane možete učitati, ili zadati kao neke konstantne vrijednosti. Zatim, pozovite funkciju `prvih_k` za  $k = 3$ . Na kraju, dealocirajte memoriju za sve elemente. Za rješavanje ovog podzadatka nije nužno riješiti podzadatke (b) i (c).
- U (b) i (c) nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 4.** (15 bodova) U tekstualnoj datoteci `kaptol.txt` nalazi se niz prirodnih brojeva manjih od 1000, odvojenih razmacima. Brojevi predstavljaju autobusne linije, poput 106. Za svaki broj  $i$  iz te datoteke, postoji datoteka `i.txt` ( $i$  je dekadski zapis broja  $i$ ), primjerice `106.txt`, u kojoj se nalazi reci sljedećeg oblika:

`sat:minute S`

Pritom je `sat` cijeli broj između 0 i 23 (uključivo), `minute` je cijeli broj između 0 i 59 (uključivo), tako da `sat:minute` označava vrijeme polaska linije, a `S` je znak 'A' ili 'B', koji označava smjer linije.

Napišite funkciju `ispis` koja prima dva cijela broja, `sat` i `minute` (u propisanom rasponu), te znak koji označava smjer kretanja ('A' ili 'B'). Označimo vrijeme određeno ulaznim argumentima funkcije s  $t$ , a vrijeme 23:59 istoga dana s  $p$ . Potrebno je pronaći **minimalno** vrijeme  $x$ , uz  $t \leq x \leq p$ , također izraženo u obliku `sat:minute`, takvo da postoji ukupno barem šest polazaka nekih linija unutar vremenskog segmenta  $[x, \min\{p, x + 5 \text{ minuta}\}]$  u zadanom smjeru ('A' ili 'B').

Ako takvo vrijeme  $x$  postoji, funkcija u datoteku `izlaz.txt` ispisuje pronađeno vrijeme  $x$  u formatu `sat:minute`. Ako je datoteka toga imena već postojala, prebrišite ju. Ako takvo vrijeme  $x$  ne postoji, ne treba ništa pisati.

**Napomene:** Osim stringova za imena datoteka, **nije** dopušteno korištenje dodatnih nizova, ali **možete** proizvoljan broj puta otvarati i zatvarati datoteke. Možete koristiti zaglavlje `string.h`. Možete pretpostaviti da sve potrebne ulazne datoteke postoje, te da su podaci unutar njih ispravno zapisani. Datoteke nisu nužno sortirane po vremenu polaska i smjeru, ali unutar pojedine datoteke nema ponavljanja redaka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Rezultati i uvidi u kolokvije:** Rezultati u četvrtak, 28.6., navečer na webu, a uvidi u petak, 29.6., u 13 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.** (12+8 bodova) Svakom stringu  $s$  pridružujemo vrijednost (tipa `double`) funkcijom  $f(s) = \sum_i \text{ascii}(s[i])/(i+2)$ , gdje `ascii` označava ASCII kôd znaka  $s[i]$ , a suma ide po svim indeksima  $i$  za koje je  $s[i] \neq '\0'$ . Niz stringova zadan je poljem pokazivača na stringove (prve znakove).

- Napišite funkciju `int evaluiraj_i_promijeni(char ** polje, int n)` koja, za ulazni niz stringova `polje` i duljinu polja `n`, računa prosječnu vrijednost funkcije  $f$  na svim stringovima iz tog polja. Za sve stringove  $s$ , za koje je  $f(s)$  **manje** ili jednako od prosjeka, treba izbaciti pripadni pokazivač iz polja `polje` (`polje` ne treba realocirati). Tako promijenjeno polje treba sortirati **silazno** prema vrijednosti funkcije  $f$ , koristeći samo zamjene pokazivača. Funkcija vraća novu duljinu polja `polje`.
- Napišite funkciju `kreiraj` koja prima jedno polje znakova, u kojem je spremljen neki (nepoznati) broj stringova, tako da su stringovi nanizani jedan za drugim (kao niz znakova), a na kraju niza nalazi se string "gotovo". Funkcija treba kreirati (alocirati) pripadno polje **pokazivača** na prvi znak svakog stringa u nizu (bez završnog stringa "gotovo"). Funkcija vraća pokazivač na prvi element tog polja (ili `NULL`, ako je polje prazno), a kroz varijabilni argument treba vratiti broj pokazivača u tom polju (tj. broj stringova u ulaznom nizu, bez završnog). Alocirajte točno onoliko memorije koliko treba za spremanje polja pokazivača. Dozvoljeno je koristiti `malloc` i `realloc`.

**Napomene:** Zabranjeno je koristiti dodatna polja, osim navedenih. Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 2.** (20 bodova) Definirajte tip podataka **napitak** kao strukturu koja sadrži sljedeće podatke: string *ime*, polje stringova *sastojci*, te *duljina* polja i *alkohol* (postotak) kao cjelobrojni tip. Duljina svih stringova je ograničena na 30 znakova.

- (a) Napišite funkciju **kreiraj**, koja alocira strukturu za novi napitak, učitava sve navedene podatke sa standardnog ulaza, i vraća pokazivač na alociranu strukturu.
- b) Napišite funkciju **barmen**, koja prima polje napitaka i sa standardnog ulaza učitava podatke “opasnog” napitka. Zatim, u polju traži napitak koje ima najmanje zajedničkih sastojaka s opasnim napitkom. Ako takvih napitaka ima više, onda se uzima onaj s najmanjim alkoholom. Ako i takvih ima više, uzima se onaj najvećeg indeksa. Na mjesto nađenog napitka (u polju) treba upisati opasni napitak, a stari (onaj nađeni) treba vratiti kroz varijabilni argument. Funkcija vraća 1 ako stari napitak nema niti jedan sastojak opasnog, a 0 inače.

**Napomene:** Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 3.** (2 + 5 + 10 + 3 bodova) Podatke o barovima u jednoj četvrti spremamo u vezanu listu. Za svaki bar pamtimo naziv (string od najviše 50 znakova), broj gostiju (prirodni broj) i promet bara (realni broj). Vezana lista barova je uvijek *silazno* sortirana po broju gostiju. Možete pretpostaviti da ne postoje dva bara s istim brojem gostiju.

- (a) Napišite definiciju odgovarajućeg tipa podatka za bar, na način da bude moguće definirati varijable naredbom `bar a;` i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
  - (b) Napišite funkciju `bar* unesi(bar* first, bar* new)` koja prima vezanu listu barova (`first` je pokazivač na početak ulazne vezane liste) te bar `new`. Funkcija ubacuje element `new` u vezanu listu tako da vezana lista i dalje bude *silazno* sortirana po broju gostiju. Ulazna vezana lista može biti prazna. Funkcija vraća pokazivač na prvi element dobivene vezane liste, nakon ubacivanja novog elementa.
  - (c) Napišite funkciju `bar* prvih_k(bar** first, int k)` koja traži prvih `k` susjednih barova s ukupnim najmanjim prometom i izbacuje ih iz vezane liste. Dva bara su susjedna ako u vezanoj listi ne postoji bar koji se nalazi između njih. Funkcija vraća pokazivač na početak vezane liste izbačenih barova (koja, također, mora biti sortirana *silazno* po broju gostiju). Ukoliko u ulaznoj vezanoj listi ima strogo manje od `k` elemenata, funkcija ne mijenja ulaznu vezanu listu i vraća `NULL`.
  - (d) U glavnom programu formirajte vezanu listu od 5 barova (elemenata), koristeći funkciju `unesi`. Podatke za te barove možete učitati, ili zadati kao neke konstantne vrijednosti. Zatim, pozovite funkciju `prvih_k` za  $k = 2$ . Na kraju, dealocirajte memoriju za sve elemente. Za rješavanje ovog podzadatka nije nužno riješiti podzadatke (b) i (c).
- U (b) i (c) nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 4.** (15 bodova) U tekstualnoj datoteci `crnomerec.txt` nalazi se niz prirodnih brojeva manjih od 1000, odvojenih razmacima. Brojevi predstavljaju autobusne linije, poput 109. Za svaki broj  $i$  iz te datoteke, postoji datoteka `i.txt` ( $i$  je dekadski zapis broja  $i$ ), primjerice `109.txt`, u kojoj se nalazi reci sljedećeg oblika:

`S sat.minute`

Pritom je `sat` cijeli broj između 0 i 23 (uključivo), `minute` je cijeli broj između 0 i 59 (uključivo), tako da `sat.minute` označava vrijeme polaska linije, a `S` je znak `'a'` ili `'b'`, koji označava smjer linije.

Napišite funkciju `ispis` koja prima dva cijela broja, `sat` i `minute` (u propisanom rasponu), te znak koji označava smjer kretanja (`'a'` ili `'b'`). Označimo vrijeme određeno ulaznim argumentima funkcije s  $t$ , a vrijeme 23.59 istoga dana s  $p$ . Potrebno je pronaći **minimalno** vrijeme  $x$ , uz  $t \leq x \leq p$ , također izraženo u obliku `sat.minute`, takvo da postoji ukupno najviše pet polazaka nekih linija unutar vremenskog segmenta  $[x, \min\{p, x + 15 \text{ minuta}\}]$  u zadanom smjeru (`'a'` ili `'b'`).

Ako takvo vrijeme  $x$  postoji, funkcija u datoteku `izlaz.txt` ispisuje pronađeno vrijeme  $x$  u formatu `sat.minute`. Ako je datoteka toga imena već postojala, prebrišite ju. Ako takvo vrijeme  $x$  ne postoji, ne treba ništa pisati.

**Napomene:** Osim stringova za imena datoteka, **nije** dopušteno korištenje dodatnih nizova, ali **možete** proizvoljan broj puta otvarati i zatvarati datoteke. Možete koristiti zaglavlje `string.h`. Možete pretpostaviti da sve potrebne ulazne datoteke postoje, te da su podaci unutar njih ispravno zapisani. Datoteke nisu nužno sortirane po vremenu polaska i smjeru, ali unutar pojedine datoteke nema ponavljanja redaka.



## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Rezultati i uvidi u kolokvije:** Rezultati u četvrtak, 28.6., navečer na webu, a uvidi u petak, 29.6., u 13 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.** (12+8 bodova) Svakom stringu  $s$  pridružujemo vrijednost (tipa `double`) funkcijom  $f(s) = \sum_i (i-1) \cdot \text{ascii}(s[i])$ , gdje `ascii` označava ASCII kôd znaka  $s[i]$ , a suma ide po svim indeksima  $i$  za koje je  $s[i] \neq '\0'$ . Niz stringova zadan je poljem pokazivača na stringove (prve znakove).

- Napišite funkciju `int evaluiraj_i_promijeni(char ** polje, int n)` koja, za ulazni niz stringova `polje` i duljinu polja `n`, računa prosječnu vrijednost funkcije  $f$  na svim stringovima iz tog polja. Za sve stringove  $s$ , za koje je  $f(s)$  strogo **veće** od prosjeka, treba izbaciti pripadni pokazivač iz polja `polje` (`polje` ne treba realocirati). Tako promijenjeno polje treba sortirati **uzlazno** prema vrijednosti funkcije  $f$ , koristeći samo zamjene pokazivača. Funkcija vraća novu duljinu polja `polje`.
- Napišite funkciju `kreiraj` koja prima jedno polje znakova, u kojem je spremljen neki (nepoznati) broj stringova, tako da su stringovi nanizani jedan za drugim (kao niz znakova), a na kraju niza nalazi se string `"stop"`. Funkcija treba kreirati (alocirati) pripadno polje **pokazivača** na prvi znak svakog stringa u nizu (bez završnog stringa `"stop"`). Funkcija vraća pokazivač na prvi element tog polja (ili `NULL`, ako je polje prazno), a kroz varijabilni argument treba vratiti broj pokazivača u tom polju (tj. broj stringova u ulaznom nizu, bez završnog). Alocirajte točno onoliko memorije koliko treba za spremanje polja pokazivača. Dozvoljeno je koristiti `malloc` i `realloc`.

**Napomene:** Zabranjeno je koristiti dodatna polja, osim navedenih. Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 2.** (20 bodova) Definirajte tip podataka `tura` kao strukturu koja sadrži sljedeće podatke: string `drzava`, polje stringova `atrakcije`, te `duljina` polja i `trajanje` kao cjelobrojni tip. Duljina svih stringova je ograničena na 70 znakova.

- (a) Napišite funkciju `kreiraj`, koja alocira strukturu za novu tura, učitava sve navedene podatke sa standardnog ulaza, i vraća pokazivač na alociranu strukturu.
- b) Napišite funkciju `vodic`, koja prima polje tura i sa standardnog ulaza učitava podatke “idealne” ture. Zatim, u polju traži tura koja ima najviše zajedničkih atrakcija s idealnom turom. Ako takvih tura ima više, onda se uzima tura najduljeg trajanja. Ako i takvih ima više, uzima se ona najvećeg indeksa. Na mjesto nađene ture (u polju) treba upisati idealnu tura, a staru (onu nađenu) treba vratiti kroz varijabilni argument. Funkcija vraća 1 ako stara tura ima sve atrakcije idealne, a 0 inače.

**Napomene:** Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 3.** (2+5+10+3 bodova) Podatke o putovanjima u turističkoj agenciji spremamo u vezanu listu. Za svako putovanje pamtimo naziv (string od najviše 50 znakova), broj prijavljenih (cijeli broj) i cijenu putovanja (realni broj). Vezana lista putovanja je uvijek *silazno* sortirana po broju prijavljenih. Možete pretpostaviti da ne postoje dva putovanja s istim brojem prijavljenih.

- (a) Napišite definiciju odgovarajućeg tipa podatka za putovanje, na način da bude moguće definirati varijable naredbom `put a`; i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
  - (b) Napišite funkciju `put* unesi(put* first, put* new)` koja prima vezanu listu putovanja (`first` je pokazivač na početak ulazne vezane liste) te putovanje `new`. Funkcija ubacuje element `new` u vezanu listu tako da vezana lista i dalje bude *silazno* sortirana po broju prijavljenih. Ulazna vezana lista može biti prazna. Funkcija vraća pokazivač na prvi element dobivene vezane liste, nakon ubacivanja novog elementa.
  - (c) Napišite funkciju `put* prvih_k(put** first, int k)` koja traži prvih `k` susjednih putovanja s ukupnom najvećom cijenom i izbacuje ih iz vezane liste. Dva putovanja su susjedna ako u vezanoj listi ne postoji putovanje koje se nalazi između njih. Funkcija vraća pokazivač na početak vezane liste izbačenih putovanja (koja, također, mora biti sortirana *silazno* po broju prijavljenih). Ukoliko u ulaznoj vezanoj listi ima strogo manje od `k` elemenata, funkcija ne mijenja ulaznu vezanu listu i vraća `NULL`.
  - (d) U glavnom programu formirajte vezanu listu od 5 putovanja (elemenata), koristeći funkciju `unesi`. Podatke za ta putovanja možete učitati, ili zadati kao neke konstantne vrijednosti. Zatim, pozovite funkciju `prvih_k` za `k = 4`. Na kraju, dealocirajte memoriju za sve elemente. Za rješavanje ovog podzadatka nije nužno riješiti podzadatke (b) i (c).
- U (b) i (c) nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 4.** (15 bodova) U tekstualnoj datoteci `dubrava.txt` nalazi se niz prirodnih brojeva manjih od 1000, odvojenih razmacima. Brojevi predstavljaju autobusne linije, poput 209. Za svaki broj  $i$  iz te datoteke, postoji datoteka `i.txt` ( $i$  je dekadski zapis broja  $i$ ), primjerice `209.txt`, u kojoj se nalazi reci sljedećeg oblika:

`sat.minute S`

Pritom je `sat` cijeli broj između 0 i 23 (uključivo), `minute` je cijeli broj između 0 i 59 (uključivo), tako da `sat.minute` označava vrijeme polaska linije, a `S` je znak 'a' ili 'b', koji označava smjer linije.

Napišite funkciju `ispis` koja prima dva cijela broja, `sat` i `minute` (u propisanom rasponu), te znak koji označava smjer kretanja ('a' ili 'b'). Označimo vrijeme određeno ulaznim argumentima funkcije s  $t$ , a vrijeme 23.59 istoga dana s  $p$ . Potrebno je pronaći **minimalno** vrijeme  $x$ , uz  $t \leq x \leq p$ , također izraženo u obliku `sat.minute`, takvo da postoji ukupno barem pet polazaka nekih linija unutar vremenskog segmenta  $[x, \min\{p, x + 10 \text{ minuta}\}]$  u zadanom smjeru ('a' ili 'b').

Ako takvo vrijeme  $x$  postoji, funkcija u datoteku `izlaz.txt` ispisuje pronađeno vrijeme  $x$  u formatu `sat.minute`. Ako je datoteka toga imena već postojala, prebrišite ju. Ako takvo vrijeme  $x$  ne postoji, ne treba ništa pisati.

**Napomene:** Osim stringova za imena datoteka, **nije** dopušteno korištenje dodatnih nizova, ali **možete** proizvoljan broj puta otvarati i zatvarati datoteke. Možete koristiti zaglavlje `string.h`. Možete pretpostaviti da sve potrebne ulazne datoteke postoje, te da su podaci unutar njih ispravno zapisani. Datoteke nisu nužno sortirane po vremenu polaska i smjeru, ali unutar pojedine datoteke nema ponavljanja redaka.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Rezultati i uvidi u kolokvije:** Rezultati u četvrtak, 28.6., navečer na webu, a uvidi u petak, 29.6., u 13 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.** (12+8 bodova) Svakom stringu  $s$  pridružujemo vrijednost (tipa `double`) funkcijom  $f(s) = \sum_i \text{ascii}(s[i])/(i+1)$ , gdje `ascii` označava ASCII kôd znaka  $s[i]$ , a suma ide po svim indeksima  $i$  za koje je  $s[i] \neq '\0'$ . Niz stringova zadan je poljem pokazivača na stringove (prve znakove).

- Napišite funkciju `int evaluiraj_i_promijeni(char ** polje, int n)` koja, za ulazni niz stringova `polje` i duljinu polja `n`, računa prosječnu vrijednost funkcije  $f$  na svim stringovima iz tog polja. Za sve stringove  $s$ , za koje je  $f(s)$  **veće** ili jednako od prosjeka, treba izbaciti pripadni pokazivač iz polja `polje` (`polje` ne treba realocirati). Tako promijenjeno polje treba sortirati **silazno** prema vrijednosti funkcije  $f$ , koristeći samo zamjene pokazivača. Funkcija vraća novu duljinu polja `polje`.
- Napišite funkciju `kreiraj` koja prima jedno polje znakova, u kojem je spremljen neki (nepoznati) broj stringova, tako da su stringovi nanizani jedan za drugim (kao niz znakova), a na kraju niza nalazi se string "`zadnji`". Funkcija treba kreirati (alocirati) pripadno polje **pokazivača** na prvi znak svakog stringa u nizu (bez završnog stringa "`zadnji`"). Funkcija vraća pokazivač na prvi element tog polja (ili `NULL`, ako je polje prazno), a kroz varijabilni argument treba vratiti broj pokazivača u tom polju (tj. broj stringova u ulaznom nizu, bez završnog). Alocirajte točno onoliko memorije koliko treba za spremanje polja pokazivača. Dozvoljeno je koristiti `malloc` i `realloc`.

**Napomene:** Zabranjeno je koristiti dodatna polja, osim navedenih. Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 2.** (20 bodova) Definirajte tip podataka `pripravak` kao strukturu koja sadrži sljedeće podatke: string *mjesavina*, polje stringova *trave*, te *duljina* polja i *tezina* kao cjelobrojni tip. Duljina svih stringova je ograničena na 50 znakova.

- (a) Napišite funkciju `kreiraj`, koja alocira strukturu za novi pripravak, učitava sve navedene podatke sa standardnog ulaza, i vraća pokazivač na alociranu strukturu.
- b) Napišite funkciju `travar`, koja prima polje pripravaka i sa standardnog ulaza učitava podatke “opasnog” pripravka. Zatim, u polju traži pripravak koji ima najmanje zajedničkih trava s opasnim pripravkom. Ako takvih pripravaka ima više, onda se uzima pripravak najmanje težine. Ako i takvih ima više, uzima se onaj najmanjeg indeksa. Na mjesto nađenog pripravka (u polju) treba upisati opasni pripravak, a stari (onaj nađeni) treba vratiti kroz varijabilni argument. Funkcija vraća 1 ako stari pripravak nema niti jednu travu opasnog, a 0 inače.

**Napomene:** Smijete koristiti biblioteke `string.h` i `stdlib.h`.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 3.** (2 + 5 + 10 + 3 bodova) Podatke o lijekovima u jednoj apoteci spremamo u vezanu listu. Za svaki lijek pamtimo naziv (string od najviše 50 znakova), broj kupaca (prirodni broj) i količinu (realni broj). Vezana lista lijekova je uvijek *uzlazno* sortirana po broju kupaca. Možete pretpostaviti da ne postoje dva lijeka s istim brojem kupaca.

- (a) Napišite definiciju odgovarajućeg tipa podatka za lijek, na način da bude moguće definirati varijable naredbom `lijek a;` i sl. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
  - (b) Napišite funkciju `lijek* unesi(lijek* first, lijek* new)` koja prima vezanu listu lijekova (`first` je pokazivač na početak ulazne vezane liste) te lijek `new`. Funkcija ubacuje element `new` u vezanu listu tako da vezana lista i dalje bude *uzlazno* sortirana po broju kupaca. Ulazna vezana lista može biti prazna. Funkcija vraća pokazivač na prvi element dobivene vezane liste, nakon ubacivanja novog elementa.
  - (c) Napišite funkciju `lijek* prvih_k(lijek** first, int k)` koja traži prvih `k` susjednih lijekova s ukupnom najmanjom količinom i izbacuje ih iz vezane liste. Dva lijeka su susjedna ako u vezanoj listi ne postoji lijek koji se nalazi između njih. Funkcija vraća pokazivač na početak vezane liste izbačenih lijekova (koja, također, mora biti sortirana *uzlazno* po broju kupaca). Ukoliko u ulaznoj vezanoj listi ima strogo manje od `k` elemenata, funkcija ne mijenja ulaznu vezanu listu i vraća `NULL`.
  - (d) U glavnom programu formirajte vezanu listu od 5 lijekova (elemenata), koristeći funkciju `unesi`. Podatke za te lijekove možete učitati, ili zadati kao neke konstantne vrijednosti. Zatim, pozovite funkciju `prvih_k` za  $k = 1$ . Na kraju, dealocirajte memoriju za sve elemente. Za rješavanje ovog podzadatka nije nužno riješiti podzadatke (b) i (c).
- U (b) i (c) nije dozvoljeno alocirati memoriju za nove elemente, već sve treba raditi s postojećim elementima.

## Programiranje 2 – drugi kolokvij, 21. 6. 2018.

**Zadatak 4.** (15 bodova) U tekstualnoj datoteci `svetice.txt` nalazi se niz prirodnih brojeva manjih od 1000, odvojenih razmacima. Brojevi predstavljaju autobusne linije, poput 228. Za svaki broj  $i$  iz te datoteke, postoji datoteka `i.txt` ( $i$  je dekadski zapis broja  $i$ ), primjerice `228.txt`, u kojoj se nalazi reci sljedećeg oblika:

S sat:minute

Pritom je `sat` cijeli broj između 0 i 23 (uključivo), `minute` je cijeli broj između 0 i 59 (uključivo), tako da `sat:minute` označava vrijeme polaska linije, a `S` je znak 'A' ili 'B', koji označava smjer linije.

Napišite funkciju `ispis` koja prima dva cijela broja, `sat` i `minute` (u propisanom rasponu), te znak koji označava smjer kretanja ('A' ili 'B'). Označimo vrijeme određeno ulaznim argumentima funkcije s  $t$ , a vrijeme 23:59 istoga dana s  $p$ . Potrebno je pronaći **minimalno** vrijeme  $x$ , uz  $t \leq x \leq p$ , također izraženo u obliku `sat:minute`, takvo da postoji ukupno najviše šest polazaka nekih linija unutar vremenskog segmenta  $[x, \min\{p, x + 20 \text{ minuta}\}]$  u zadanom smjeru ('A' ili 'B').

Ako takvo vrijeme  $x$  postoji, funkcija u datoteku `izlaz.txt` ispisuje pronađeno vrijeme  $x$  u formatu `sat:minute`. Ako je datoteka toga imena već postojala, prebrišite ju. Ako takvo vrijeme  $x$  ne postoji, ne treba ništa pisati.

**Napomene:** Osim stringova za imena datoteka, **nije** dopušteno korištenje dodatnih nizova, ali **možete** proizvoljan broj puta otvarati i zatvarati datoteke. Možete koristiti zaglavlje `string.h`. Možete pretpostaviti da sve potrebne ulazne datoteke postoje, te da su podaci unutar njih ispravno zapisani. Datoteke nisu nužno sortirane po vremenu polaska i smjeru, ali unutar pojedine datoteke nema ponavljanja redaka.