JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! Mobitele isključite i spremite! 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.

Napomena: Svi zadaci su programski zadaci, u smislu uvjeta polaganja kolegija (80% bodova na barem jednom zadatku).

Zadatak 1 (15=8+5+2 bodova) U ovom zadatku **ne smijete** koristiti nizove, ali smijete pisati pomoćne funkcije.

a) Napišite funkcije `transformacija1` i `transformacija2` koje preko varijabilnih argumenata primaju dva cijela broja a i b te mijenjaju njihove vrijednosti na sljedeći način: `transformacija1` mijenja (a, b) u $(c, \text{GCD}(a, b))$, gdje je c najveći prosti faktor od $|a|$ (ako takav ne postoji, uzimamo $c = 1$), dok `transformacija2` mijenja (a, b) u $(2 \cdot |a - 6b|,\ 4 \cdot |a + b|)$.

b) Napišite funkciju `promjena` koja prima dva cijela broja a i b (po vrijednosti) i, po potrebi, neke dodatne parametre. Funkcija učitava od korisnika cijele brojeve sve dok ne učita broj različit od 1 i 2 te redom vrši niz transformacija nad (a, b) na sljedeći način: kada učitamo 1 poziva se funkcija `transformacija1`, a kada učitamo 2 poziva se `transformacija2` (vidi primjer). Koristite funkcije iz a), čak i ako ih niste napisali. Funkcija mora vratiti nove vrijednosti (a, b) preko varijabilnih argumenata, a kao povratnu vrijednost vraća ukupan broj provedenih transformacija.

c) Napišite program koji od korisnika učitava dva cijela broja a i b te poziva funkciju `promjena`. Program treba ispisati početnu vrijednost od (a, b), novu vrijednost nakon poziva funkcije te ukupan broj provedenih transformacija.

Primjer: Za $a = 145$ i $b = 15$ i učitane brojeve 1 2 1 0 program treba ispisati 145 15 2 2 3 jer imamo sljedeće transformacije:

$$(145, 15) \xrightarrow{1} (29, 5) \xrightarrow{2} (2, 136) \xrightarrow{1} (2, 2).$$

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Zadatak 2 (20=8+12 bodova)

a) Napišite funkciju `double horner(double a[], int n, double x)` koja za polje koeficijenata `a` veličine `n+1` i realan broj `x` Hornerovim algoritmom računa $p(x)$ za polinom:

$$p(x) = \sum_{i=0}^{n} \left(\sum_{j=0}^{n-i} a_{n-j} \right) x^{2n-2i+1}.$$

b) Napišite funkciju `void sort(double a[], int n, double x, int b[], int m)` koja prima `a`, `n`, `x` kao iz a) te strogo rastući (ne treba provjeravati) niz nenegativnih cijelih brojeva `b` duljine `m`. Funkcija treba sortirati niz `b` **uzlazno** po sljedećem uređaju: broj k je manji od broja l ukoliko vrijedi $p^{(k)}(x) < p^{(l)}(x)$ ili ako je $p^{(k)}(x) = p^{(l)}(x)$ i $k < l$, pri čemu je:

$$p^{(k)}(x) = \underbrace{(p \circ p \circ \ldots \circ p)}_{k}(x) \quad \text{i} \quad p^{(0)}(x) = x$$

za polinom p definiran u a). Smijete koristiti najviše jedno pomoćno polje veličine `m` u koje ćete spremati rezultate evaluacije polinoma. Rješenje podzadatka b) koje poziva funkciju `horner` više od `b[m-1]` puta nosi najviše 10 bodova. Također, smijete koristiti funkciju `horner` bez da ste ju napisali.

Primjer: Za $a = (-3, 1)$ i $x = 1$ funkcija `sort` mora promijeniti niz $b = (1, 2, 3)$ u $(1, 3, 2)$ jer polinom u ovom slučaju glasi $p(x) = -2x^3 + x$ te vrijedi $p^{(1)}(1) = -1$, $p^{(2)}(1) = p(p(1)) = 1$ i $p^{(3)}(1) = p(p(p(1))) = -1$.

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Zadatak 3 (15 bodova) Napišite funkciju `del_zn` koja iz niza cijelih brojeva (gdje je niz duljine $n > 0$), briše sve elemente koji u zapisu u bazi 10 imaju više od 3 parne znamenke. Funkcija kao povratnu vrijednost vraća novu duljinu niza, a kroz varijabilne argumente vraća minimalni i maksimalni element promijenjenog niza. Prototip funkcije potreban za rješenje zadatka osmislite sami.

Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**.

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Zadatak 4 (20=8+12 bodova) Pokemon treneri Misty i Ash stoje na stazi na kojoj se nalaze pokemoni i natječu tko će ih uloviti više. Na stazi je i sudac natjecanja Brock. Trenutno stanje staze dano je nizom `S` tipa `char`. Duljina niza `S` je `n`. Položaj pokemon trenera na stazi određen je slovima `'m'`, odnosno `'a'`, a položaj suca određen je slovom `'b'`. Mjesta na kojima se na stazi nalaze pokemoni u nizu su označena slovom `'p'`, dok su prazna mjesta na stazi u nizu označena znakom `'*'`. Misty i Ash naizmjenično bacaju Poké lopte. Ako trener baci loptu na mjesto na kojem je pokemon, on je uhvatio pokemona i mjesto postaje prazno. Jednom bačena lopta ne može se više koristiti. Natjecanje je završeno kad oba trenera ostanu bez lopti, kad na stazi više nema pokemona ili ako lopta pogodi suca. Pobjednik natjecanja je trener koji je ulovio više pokemona. Ako su oba trenera ulovila isti broj pokemona, pobjednika nema. Ako je natjecanje završeno jer je pogođen sudac, trener koji je pogodio suca je gubitnik bez obzira na broj ulovljenih pokemona.

a) Napišite funkciju `int bacanjePokeLopte(char S[], int n, char t, int k)` koja simulira bacanje Poké lopte od strane trenera s oznakom `t`. Ako je `k` < 0, lopta pada $|$`k`$|$ mjesta lijevo od trenera, u suprotnom pada $|$`k`$|$ mjesta desno. Ako lopta odleti izvan staze, pomiče se na kraj, odnosno na početak staze. Funkcija vraća 1 ako je trener `t` uhvatio pokemona, -1 ako je lopta pogodila suca, te 0 u ostalim slučajevima.

b) Napišite funkciju `int natjecanje(char S[], int n, int brP, char *t)` koja simulira natjecanje. Svaki trener na početku ima `brP` Poké lopti. Oznaka trenera koji prvi baca loptu zapisana je na adresi `t`. Funkcija vraća koliko je pokemona uhvatio pobjednik te oznaku trenera koji je pobijedio (preko varijabilnog parametra `t`). Ako nema pobjednika, funkcija vraća -1, a varijabilni argument `t` postavlja se na `'*'`. Smijete koristiti funkciju iz a) zadatka bez da ste ju napisali. Za određivanje "putanje" Poké lopte koristite funkciju `letiLetiLopta(int r)` koja vraća nasumičan cijeli broj iz segmenta $[-r, r]$ (pretpostavite da je funkcija već napisana). Pretpostavljamo da se kod poziva ove funkcije na stazi nalazi barem jedan pokemon.

Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**.

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Upute: Na kolokviju je dozvoljeno koristiti samo pribor za pisanje i brisanje. Kalkulatori, razne neslužbene tablice, papiri i sl., nisu dozvoljeni! Mobitele isključite i spremite! 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.

Napomena: Svi zadaci su programski zadaci, u smislu uvjeta polaganja kolegija (80% bodova na barem jednom zadatku).

Zadatak 1 (15=8+5+2 bodova) U ovom zadatku **ne smijete** koristiti nizove, ali smijete pisati pomoćne funkcije.

a) Napišite funkcije `promjenaA` i `promjenaB` koje preko varijabilnih argumenata primaju dva cijela broja x i y te mijenjaju njihove vrijednosti na sljedeći način: `promjenaA` mijenja (x, y) u $(2 \cdot |x + y|,\ 3 \cdot |x - 3y|)$, dok `promjenaB` mijenja (x, y) u $(\mathrm{GCD}(x, y),\ z)$, gdje je z broj različitih prostih faktora od $|y|$ (ako ne postoje ili ako je $y = 0$ uzimamo $z = 0$).

b) Napišite funkciju `transformacija` koja prima dva cijela broja x i y (po vrijednosti) i, po potrebi, neke dodatne parametre. Funkcija učitava od korisnika znakove sve dok ne učita znak različit od 'A' i 'B' te redom vrši niz transformacija nad (x, y) na sljedeći način: kada učitamo 'A' poziva se funkcija `promjenaA`, a kada učitamo 'B' poziva se `promjenaB` (vidi primjer). Koristite funkcije iz a), čak i ako ih niste napisali. Funkcija mora vratiti nove vrijednosti (x, y) preko varijabilnih argumenata, a kao povratnu vrijednost vraća ukupan broj učitanih znakova.

c) Napišite program koji od korisnika učitava dva cijela broja x i y te poziva funkciju `transformacija`. Program treba ispisati početnu vrijednost od (x, y), novu vrijednost nakon poziva funkcije te ukupan broj učitanih znakova.

Primjer: Za $x = 145$ i $y = 15$ i učitane znakove $BABC$ program treba ispisati 145 15 1 1 4 jer imamo sljedeće promjene:

$$(145, 15) \xrightarrow{B} (5, 2) \xrightarrow{A} (14, 3) \xrightarrow{B} (1, 1).$$

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Zadatak 2 (20=8+12 bodova)

a) Napišite funkciju `double horner(double c[], int n, double x)` koja za polje koeficijenata `c` veličine `n+1` i realan broj `x` Hornerovim algoritmom računa $q(x)$ za polinom:

$$q(x) = \sum_{i=0}^{n} \left(\sum_{j=i}^{n} c_j \right) x^{3n-3i+1}.$$

b) Napišite funkciju `void sort(double c[], int n, double x, int d[], int m)` koja prima `c`, `n`, `x` kao iz a) te strogo rastući (ne treba provjeravati) niz nenegativnih cijelih brojeva `d` duljine `m`. Funkcija treba sortirati niz `d` **silazno** po sljedećem uređaju: broj k je veći od broja l ukoliko vrijedi $q^{(k)}(x) > q^{(l)}(x)$ ili ako je $q^{(k)}(x) = q^{(l)}(x)$ i $k > l$, pri čemu je:

$$q^{(k)}(x) = \underbrace{(q \circ q \circ \ldots \circ q)}_{k}(x) \quad \text{i} \quad q^{(0)}(x) = x$$

za polinom q definiran u a). Smijete koristiti najviše jedno pomoćno polje veličine `m` u koje ćete spremati rezultate evaluacije polinoma. Rješenje podzadatka b) koje poziva funkciju `horner` više od `d[m-1]` puta nosi najviše 10 bodova. Također, smijete koristiti funkciju `horner` bez da ste ju napisali.

Primjer: Za $c = (1, -1)$ i $x = 1$ funkcija `sort` mora promijeniti niz $d = (1, 2, 3)$ u $(2, 3, 1)$ jer polinom u ovom slučaju glasi $q(x) = -x$ te vrijedi $q^{(1)}(1) = -1$, $q^{(2)}(1) = q(q(1)) = 1$ i $q^{(3)}(1) = q(q(q(1))) = -1$.

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Zadatak 3 (15 bodova) Napišite funkciju `del_zn` koja iz niza cijelih brojeva (gdje je niz duljine $n > 0$), briše sve elemente koji u zapisu u bazi 10 imaju manje od 2 neparne znamenke. Funkcija kao povratnu vrijednost vraća novu duljinu niza, a kroz varijabilne argumente vraća minimalni i maksimalni element promijenjenog niza. Prototip funkcije potreban za rješenje zadatka osmislite sami.

Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**.

JMBAG

IME I PREZIME

Programiranje 1 – drugi kolokvij, 10. 02. 2023.

Zadatak 4 (20=8+12 bodova) Lovci na meduze Spužvabob i Patrik stoje u dolini meduza i natječu tko će ih uloviti više. U dolini je i sudac natjecanja Kalamarko. Trenutno stanje doline dano je nizom `D` tipa `char`. Duljina niza `D` je `n`. Položaj lovaca u dolini određen je slovima `'s'`, odnosno `'p'`, a položaj suca određen je slovom `'k'`. Mjesta na kojima se u dolini nalaze meduze u nizu su označena slovom `'m'`, dok su prazna mjesta u dolini u nizu označena znakom `'*'`. Spužvabob i Patrik naizmjenično bacaju mreže. Ako lovac baci mrežu na mjesto na kojem je meduza, on je uhvatio meduzu i mjesto postaje prazno. Jednom bačena mreža ne može se više koristiti. Natjecanje je završeno kad oba trenera ostanu bez mreža, kad u dolini više nema meduza ili ako mreža pogodi suca. Pobjednik natjecanja je lovac koji je ulovio više medzua. Ako su oba lovca ulovila isti broj meduza, pobjednika nema. Ako je natjecanje završeno jer je pogođen sudac, lovac koji je pogodio suca je gubitnik bez obzira na broj ulovljenih meduza.

a) Napišite funkciju `int bacanjeMreze(char D[], int n, char l, int a)` koja simulira bacanje mreže od strane lovca s oznakom `l`. Ako je `a` < 0, mreža pada $|$`a`$|$ mjesta desno od lovca, u suprotnom pada $|$`a`$|$ mjesta lijevo. Ako mreža odleti izvan doline, pomiče se na kraj, odnosno na početak doline. Funkcija vraća 1 ako je lovac `l` uhvatio meduzu, -1 ako je mreža pogodila suca, te 0 u ostalim slučajevima.

b) Napišite funkciju `int natjecanje(char D[], int n, int brM, char *l)` koja simulira natjecanje. Svaki lovac na početku ima `brM` mreža. Oznaka lovca koji prvi baca mrežu zapisana je na adresi `l`. Funkcija vraća koliko je meduza uhvatio pobjednik te oznaku lovca koji je pobijedio (preko varijabilnog parametra `l`). Ako nema pobjednika, funkcija vraća -1, a varijabilni argument `l` postavlja se na `'*'`. Smijete koristiti funkciju iz a) zadatka bez da ste ju napisali. Za određivanje "putanje" mreže koristite funkciju `letiLetiMreza(int r)` koja vraća nasumičan cijeli broj iz segmenta $[-r, r]$ (pretpostavite da je funkcija već napisana). Pretpostavljamo da se kod poziva ove funkcije u dolini nalazi barem jedna meduza.

Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**.