

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Rezultati i uvidi: Rezultati do ponedjeljka 12.9.2022. navečer na webu. Termini uvida bit će objavljeni naknadno.

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!** 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. Uvjete navedene u zadacima (nenegativnost, ograde na n i sl.) ne treba provjeravati.

Zadatak 1 ($7 + 18 = 25$ bodova) Bob Graditelj želi izgraditi ogradu za svoju kuću. Bob ima na raspolaganju jednu veliku dasku duljine n decimetara, iz koje planira ispiliti daske za ogradu (uz mogući višak materijala). Ograda se treba sastojati od točno k vertikalnih dasaka poredanih u niz. Pritom trebaju biti zadovoljeni sljedeći uvjeti:

- Duljina svake daske u decimetrima prirodan je broj.
- Duljina svake daske osim prve i posljednje veća je od ili jednaka aritmetičkoj sredini duljina susjednih dasaka.
- Bob je praznovjeran pa smatra da duljine svih dasaka u decimetrima ne smiju imati zajednički djelitelj veći od jedan.

Boba zanima koliko različitih ograda može izgraditi poštujući navedene zahtjeve. Na primjer, ako je $n = 5$ i $k = 3$, tada postoji pet mogućih ograda: $(1, 1, 1)$, $(1, 2, 1)$, $(1, 3, 1)$, $(1, 2, 2)$, $(2, 2, 1)$.

- (a) Napišite rekurzivnu funkciju `nzd` koja prima dva prirodna broja te vraća njihov najveći zajednički djelitelj (*uputa*: koristite Euklidov algoritam).
- (b) Napišite rekurzivnu funkciju `ograde` koja prima prirodne brojeve n i k iz teksta zadatka (te po potrebi još neke argumente) te vraća broj ograda koje Bob može izgraditi u skladu s navedenim uvjetima. Napišite i primjer poziva funkcije unutar glavnog programa.

Napomena. Možete pretpostaviti da vrijedi $3 \leq k \leq n \leq 100$ te da broj mogućih ograda stane u tip podataka `int`. U ovom zadatku smijete koristiti samo biblioteku `stdio.h` te ne smijete koristiti nizove.

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Zadatak 2 (5 + 10 + 15 = 30 bodova)

- (a) Napišite funkciju `char* ucitaj_rijec(...)` (sami odredite argumente te funkcije tako da ju možete koristiti u idućem podzadatku ovog zadatka) koja omogućava da se od korisnika učita jedna riječ (riječ je neprazan niz velikih ili malih slova, a učitavanje završava unosom znaka koji nije slovo) i koja vraća pokazivač na alocirani string u kojem je ta riječ spremljena.
- (b) Napišite funkciju `void unos(int* duljina, char*** rijeci, int** br_pojava)` koja od korisnika prvo učita prirodan broj n . Funkcija zatim učitava riječi koje su međusobno odvojene bar jednim znakom koji nije slovo, sve dok ne učita ukupno n znakova točka (tj. dok ne učita n rečenica).

Svaku riječ koju učitamo (za učitavanje riječi možete koristiti funkciju iz (a) podzadatka) potrebno je spremiti u niz `*rijeci`. Pritom u nizu ne smiju biti spremljene iste riječi više puta (pri čemu ne razlikujemo velika i mala slova, npr. "Euforija" je za nas isto kao "euFORija"), te broj (`*br_pojava`)[i] mora predstavljati koliko se puta riječ (`*rijeci`)[i] pojavila prilikom unosa (pri čemu ponovo ne razlikujemo velika i mala slova). Ovdje `*duljina` predstavlja broj riječi koje se nakon učitavanja nalaze u nizu `*rijeci`. Također, nije dozvoljeno pretpostavljati početne vrijednosti argumenata ove funkcije prilikom njezina poziva.

- (c) Napišite glavni program (tj. funkciju `main`) koji poziva funkciju iz (b) podzadatka. Dobiveni niz riječi zatim treba sortirati silazno po abecedi, pri čemu ne razlikujemo velika i mala slova. Zatim treba ispisati te riječi zajedno s brojem njihova pojavljivanja u unosu (pogledajte primjer).

Za sortiranje je potrebno koristiti Quicksort algoritam. Rješenje koje koristi neki drugi način sortiranja nosi najviše 5 bodova (za ovaj podzadatak).

Primjer.

Unos	Ispis*	
3	vezda 1	nebi 2
Nigdar ni tak bilo	tak 2	nam 1
da ni nekak bilo,	pak 1	kak 2
pak Ni vezda nebu	nikak 1	Kajti 1
da nam neKak nebu..	Nigdar 1	je 1
Kajti: kak bi bilo	ni 4	da 3
da nebi nEkaK bilo,	nekak 3	bilo 6
nebi biLo nikak,	nebu 2	bi 1
ni tAk kak je bilo.		

(*Ispis ne mora biti u dva stupca!)

Napomena. Svi stringovi i polja koja se koriste u zadatku moraju biti dinamički (re)alocirani (i naposljetku dealocirani), te zauzimati točno koliko je memorije potrebno. U ovom zadatku dozvoljeno je korištenje sljedećih biblioteka: `stdio.h`, `stdlib.h`, `string.h`, `cctype.h`.

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Zadatak 3 (2+7+5+6+10 = 30 bodova) Skup kompleksnih brojeva spremamo u vezanu listu. Za svaki kompleksni broj pamtimo njegov realni i imaginarni dio. Smatramo da su dva kompleksna broja jednaka ukoliko imaju isti realni i imaginarni dio. Pretpostavljamo da su u skupu svi elementi različiti.

- Napišite deklaraciju odgovarajućeg tipa podataka za kompleksni broj, na način da bude moguće definirati varijablu naredbom `complex a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- Napišite funkciju `complex* dodaj_sort(complex *novi, complex *first)` koja prima pokazivač `novi` na novi kompleksni broj i pokazivač `first` na listu kompleksnih brojeva koja je ili prazna ili sortirana uzlazno po modulu kompleksnog broja. Funkcija treba dodati kompleksni broj `novi` u listu `first` tako da ona ostane uzlazno sortirana po modulu kompleksnog broja. Ako dva kompleksna broja imaju isti modul, njihov poredak nije važan. Funkcija vraća pokazivač na prvi element ažurirane liste.
- Napišite funkciju `complex* izbaci(complex *first, double x, double y)` koja prima pokazivač `first` na listu kompleksnih brojeva. Funkcija pronalazi element s realnim dijelom `x` i imaginarnim dijelom `y` i izbacuje ga iz liste. Funkcija vraća pokazivač na prvi element ažurirane liste.
- Napišite funkciju `complex* okreni(complex* first)` koja prima pokazivač `first` na listu kompleksnih brojeva. Funkcija treba obrnuti elemente liste.
- Koristeći funkcije iz podzadataka b, c i d (čak i ako ih niste napisali), napišite funkciju `complex* razlika(complex **A, complex **B)`. Funkcija prima pokazivače na dva pokazivača na liste kompleksnih brojeva koje su sortirane uzlazno po modulu. Funkcija stvara novu listu kompleksnih brojeva koja predstavlja razliku skupova $A \setminus B$ i vraća pokazivač na njen prvi element. Lista mora biti sortirana uzlazno po modulu. Dodatno, preko varijabilnih argumenata vraća pokazivač na prvi element promijenjenih listi `A` i `B`. U listi `A` ostaju samo elementi iz presjeka $A \cap B$, a u listi `B` elementi razlike $B \setminus A$. Dodatno, skupovi `A` i `B` trebaju biti sortirani silazno po modulu.

U zadatku nije dozvoljeno korištenje dodatne memorije te pomoćnih listi (osim rezultatne liste u podzadatku e).

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Zadatak 4 (30 bodova) Podaci o jednom artiklu u prodavaonici su spremljeni kao struktura naziva `artikal` koja se sastoji od naziva artikla (string duljine najviše 50 znakova), šifre artikla (cijeli broj) i stanja na skladištu (cijeli broj).

Napišite program koji kao argumente komandne linije prima nazive jedne binarne i jedne tekstualne datoteke. U binarnoj datoteci je zapisano trenutno stanje artikala u prodavaonici.

Pretpostavite da imate već učitani cijeli broj `n` i matricu cijelih brojeva `kosarica` s dva stupca i `n` redaka. Prvi stupac matrice `kosarica` predstavlja šifre artikala koje kupac želi kupiti, a drugi stupac predstavlja koliko tih artikala želi kupiti. Jedan artikal se može pojaviti više puta u matrici.

Program u tekstualnoj datoteci zapisuje artikle koje je moguće kupiti na sljedeći način:

- Ukoliko artikla ima dovoljno na trenutnom stanju u datoteku zapišemo jedan ispod drugog: naziv, šifru i količinu kupljenih artikala;
- Ukoliko artikla na stanju ima, ali ne dovoljno, zapisujemo jedan ispod drugog: naziv, šifru i broj artikala na stanju sa dodanim znakom `'*'`;
- Ukoliko artikla nema na stanju, artikal se ne zapisuje u tekstualnu datoteku.

Binarna datoteka se ažurira tako da na kraju ima zapisano novo stanje u prodavaonici, nakon što se obavi kupnja određena matricom `kosarica`.

Primjer:

```
ulazna bin dat: kosarica: izlazna txt dat.: ažurirana bin dat.:
Ormar           2341 1   Stolica        Ormar
1234            3412 4   2341          1234
2               2341 1   2             2
Stolica         Umivaonik  Umivaonik    Stolica
2341            3412          2341
5               3*          3
Umivaonik      Umivaonik    Umivaonik
3412            3412          3412
3               0             0
```

Za svaku datoteku provjerite uspješnost njenog otvaranja te ih zatvorite nakon korištenja.

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Rezultati i uvidi: Rezultati do ponedjeljka 12.9.2022. navečer na webu. Termini uvida bit će objavljeni naknadno.

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!** 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. Uvjete navedene u zadacima (nenegativnost, ograde na n i sl.) ne treba provjeravati.

Zadatak 1 ($7 + 18 = 25$ bodova) Pat i Mat žele izgraditi ogradu za svoju kuću. Oni imaju na raspolaganju jednu veliku dasku duljine n decimetara, iz koje planiraju ispiliti daske za ogradu. Ograda se treba sastojati od točno k vertikalnih dasaka poredanih u niz. Pritom trebaju biti zadovoljeni sljedeći uvjeti:

- Duljina svake daske u decimetrima prirodan je broj.
- Duljina svake daske osim prve i posljednje manja je od ili jednaka aritmetičkoj sredini duljina susjednih dasaka.
- Iskorišten je sav materijal, odnosno zbroj duljina svih dasaka jednak je točno n .

Dodatno, Pat i Mat definiraju *ljepotu* neke ograde kao najveći zajednički djelitelj duljina dasaka u njoj. Njih zanima koliki je zbroj ljepota svih ograda koje zadovoljavaju navedene uvjete. Na primjer, ako je $n = 6$ i $k = 3$, tada postoji sedam mogućih ograda: $(1, 1, 4)$, $(2, 1, 3)$, $(3, 1, 2)$, $(4, 1, 1)$, $(1, 2, 3)$, $(2, 2, 2)$, $(3, 2, 1)$, stoga je traženi zbroj ljepota $1 + 1 + 1 + 1 + 1 + 2 + 1 = 8$.

- Napišite rekurzivnu funkciju `nzd` koja prima dva prirodna broja te vraća njihov najveći zajednički djelitelj (*uputa*: koristite Euklidov algoritam).
- Napišite rekurzivnu funkciju `zbroj` koja prima prirodne brojeve n i k iz teksta zadatka (te po potrebi još neke argumente) te vraća zbroj ljepota svih ograda koje Pat i Mat mogu izgraditi u skladu s navedenim uvjetima. Napišite i primjer poziva funkcije unutar glavnog programa.

Napomena. Možete pretpostaviti da vrijedi $3 \leq k \leq n \leq 100$ te da zbroj ljepota svih mogućih ograda stane u tip podataka `int`. U ovom zadatku smijete koristiti samo biblioteku `stdio.h` te ne smijete koristiti nizove.

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Zadatak 2 (5 + 10 + 15 = 30 bodova)

- (a) Napišite funkciju `char* spremi_rijec(...)` (sami odredite argumente te funkcije tako da ju možete koristiti u idućem podzadatku ovog zadatka) koja omogućava da se od korisnika učita jedna riječ (riječ je neprazan niz znamenaka, velikih ili malih slova, poput "123", "ab2" ili "abCd", a učitavanje završava unosom znaka koji nije ni slovo ni znamenka) i koja vraća pokazivač na alocirani string u kojem je ta riječ spremljena.
- (b) Napišite funkciju `void obradi_tekst(char*** rijeci, int** br_pojava, int* duljina)` koja od korisnika prvo učita prirodan broj m . Funkcija zatim učitava riječi koje su međusobno odvojene bar jednim znakom koji nije ni slovo ni znamenka, sve dok ne učita ukupno m znakova točka ili zarez (tj. '.' ili ',').

Svaku riječ koju učitamo (za učitavanje riječi možete koristiti funkciju iz (a) podzadatka) potrebno je spremiti u niz `*rijeci`. Pritom u nizu ne smiju biti spremljene iste riječi više puta (pri čemu ne razlikujemo velika i mala slova, npr. "Euforija123" je za nas isto kao "euFORiJA123"), te broj (`*br_pojava`) [i] mora predstavljati koliko se puta riječ (`*rijeci`) [i] pojavila prilikom unosa (pri čemu ponovo ne razlikujemo velika i mala slova). Ovdje `*duljina` predstavlja broj riječi koje se nakon učitavanja nalaze u nizu `*rijeci`. Također, nije dozvoljeno pretpostavljati početne vrijednosti argumenata ove funkcije prilikom njezina poziva.

- (c) Napišite glavni program (tj. funkciju `main`) koji poziva funkciju iz (b) podzadatka. Dobiveni niz riječi zatim treba sortirati uzlazno, pri čemu ne razlikujemo velika i mala slova. Zatim treba ispisati te riječi zajedno s brojem njihova pojavljivanja u unosu (pogledajte primjer).

Za sortiranje je potrebno koristiti Quicksort algoritam. Rješenje koje koristi neki drugi način sortiranja nosi najviše 5 bodova (za ovaj podzadatak).

Primjer.

Unos	Ispis*	
6	2a 1	nebu 2
Nigdar n1 tak bilo	bi 1	nebi 2
2a n1 nekak bilo,	bilo 6	nekak 3
pak N1 vezda nebu	da 2	Nigdar 1
da nam neKak nebu..	je 1	nikak 1
Kajti: kak bi bilo	Kajti 1	pak 1
da nebi nEkaK bilo,	kak 2	tak 2
nebi biLo nikak,	n1 4	vezda 1
n1 tAk kak je bilo.	nam 1	

(*Ispis ne mora biti u dva stupca!)

Napomena. Svi stringovi i polja koja se koriste u zadatku moraju biti dinamički (re)alocirani (i naposljetku dealocirani), te zauzimati točno koliko je memorije potrebno. U ovom zadatku dozvoljeno je korištenje sljedećih biblioteka: `stdio.h`, `stdlib.h`, `string.h`, `ctype.h`.

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Zadatak 3 (2+7+5+6+10 = 30 bodova) Skup kompleksnih brojeva spremamo u vezanu listu. Za svaki kompleksni broj pamtimo njegov realni i imaginarni dio. Smatramo da su dva kompleksna broja jednaka ukoliko imaju isti realni i imaginarni dio. Pretpostavljamo da su u skupu svi elementi različiti.

- Napišite deklaraciju odgovarajućeg tipa podataka za kompleksni broj, na način da bude moguće definirati varijablu naredbom `complex a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- Napišite funkciju `complex* dodaj_sort(complex *novi, complex *first)` koja prima pokazivač `novi` na novi kompleksni broj i pokazivač `first` na listu kompleksnih brojeva koja je ili prazna ili sortirana silazno po modulu kompleksnog broja. Funkcija treba dodati kompleksni broj `novi` u listu `first` tako da ona ostane sortirana silazno po modulu kompleksnog broja. Ako dva kompleksna broja imaju isti modul, njihov poredak nije važan. Funkcija vraća pokazivač na prvi element ažurirane liste.
- Napišite funkciju `complex* izbaci(complex *first, double x, double y)` koja prima pokazivač `first` na listu kompleksnih brojeva. Funkcija pronalazi element s realnim dijelom `x` i imaginarnim dijelom `y` i izbacuje ga iz liste. Funkcija vraća pokazivač na prvi element ažurirane liste.
- Napišite funkciju `complex* okreni(complex* first)` koja prima pokazivač `first` na listu kompleksnih brojeva. Funkcija treba obrnuti elemente liste.
- Koristeći funkcije iz podzadataka b, c i d (čak i ako ih niste napisali), napišite funkciju `complex** razlika(complex**A, complex**B)`. Funkcija prima pokazivače na dva pokazivača na liste kompleksnih brojeva koje su sortirane silazno po modulu. Funkcija stvara novu listu kompleksnih brojeva koja predstavlja razliku skupova $B \setminus A$ i vraća pokazivač na njen prvi element. Lista mora biti sortirana silazno po modulu. Dodatno, preko varijabilnih argumenata vraća pokazivač na prvi element promijenjenih listi `A` i `B`. U listi `A` ostaju samo elementi razlike $A \setminus B$, a u listi `B` elementi presjeka $A \cap B$. Dodatno, skupovi `A` i `B` trebaju biti sortirani uzlazno po modulu.

U zadatku nije dozvoljeno korištenje dodatne memorije te pomoćnih listi (osim rezultatne liste u podzadatku e).

Programiranje 2 – popravni kolokvij, 5. 9. 2022.

Zadatak 4 (30 bodova) Podaci o jednom artiklu u prodavaonici su spremljeni kao struktura naziva `artikal` koja se sastoji od naziva artikla (string duljine najviše 50 znakova), šifre artikla (cijeli broj) i stanja na skladištu (nenegativni cijeli broj).

Napišite program koji kao argumente komandne linije prima nazive jedne binarne i jedne tekstualne datoteke. U binarnoj datoteci je zapisano trenutno stanje artikala u prodavaonici.

Pretpostavite da imate već učitani cijeli broj `n` i matricu cijelih brojeva `kosarica` s dva stupca i `n` redaka. Prvi stupac matrice `kosarica` predstavlja koliko se određenog artikla želi kupiti, a drugi stupac predstavlja šifru odgovarajućeg artikla. Jedan artikal se može pojaviti više puta u matrici.

Program u tekstualnoj datoteci zapisuje artikle koje je moguće kupiti na sljedeći način:

- Ukoliko artikla ima dovoljno na trenutnom stanju u datoteku zapišemo jedan ispod drugog: šifru, naziv i količinu kupljenih artikala;
- Ukoliko artikla na stanju ima, ali ne dovoljno, zapisujemo jedan ispod drugog: naziv, šifru i broj artikala na stanju sa dodanim znakom '+';
- Ukoliko artikla nema na stanju, artikal se ne zapisuje u tekstualnu datoteku.

Binarna datoteka se ažurira tako da na kraju ima zapisano novo stanje u prodavaonici, nakon što se obavi kupnja određena matricom `kosarica`.

Primjer:

```
ulazna bin dat: kosarica: izlazna txt dat.: ažurirana bin dat.:
Ormar          1 2341 2341          Ormar
1234           4 3412 Stolica          1234
2              1 2341 2              2
Stolica                3412          Stolica
2341                Umivaonik        2341
5                  3+              3
Umivaonik                Umivaonik
3412                3412
3                    0
```

Za svaku datoteku provjerite uspješnost njenog otvaranja te ih zatvorite nakon korištenja.