

## Programiranje 2 – druga popravna provjera znanja, 18. 9. 2020.

**Rezultati i uvidi:** Rezultati u srijedu, 23. 9. 2020., navečer na webu, a uvidi u četvrtak, 24. 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) Napišite rekurzivnu funkciju koja prima  $n \in \mathbb{N}$  (i eventualno još neke parametre) i vraća broj načina na koji se ploča dimenzije  $1 \times n$  može popločati pomoću pločica dimenzije  $1 \times 1$  i  $1 \times 3$ . Obavezno napišite poziv funkcije.

## Programiranje 2 – druga popravna provjera znanja, 18. 9. 2020.

**Zadatak 2.** (30 bodova) Popularna kartaška igra *The Mind* iz 2018. igra se pomoću špila od 100 karata. Na svakoj karti je jedan prirodan broj od 1 do 100 te su svi ti brojevi zastupljeni. Ovdje ćemo za potrebe zadatka neznatno pojednostaviti pravila igre.

Igra je kooperativna, odnosno svi igrači igraju protiv igre — ili pobjeđuju zajedno ili gube zajedno. Potrebno je „preživjeti” osam razina. Svaka razina  $n \in \{1, \dots, 8\}$  počinje tako da svaki od četvero igrača dobije  $n$  karata. Karte je potrebno licem prema gore spustiti na hrpu na sredini stola (koja je u početku prazna) tako da svih  $4n$  karata završi na hrpi uzlazno po veličini (karta s najmanjim brojem na dnu hrpe, s najvećim na vrhu hrpe). Nema uobičajenih poteza, nego svatko može spustiti bilo koju svoju kartu u bilo kojem trenutku. Bit igre je u tome da nije dozvoljena verbalna komunikacija, samo minimalna neverbalna komunikacija (igra dakako nije idealna za matematičare koji brzo smisle način kako zaobići to „pravilo”).

Napišite deklaraciju tipa `lista` za vezanu listu cijelih brojeva. Zatim napišite funkciju `lista rijesi(int n, lista *l)` koja prima broj  $n$  trenutne razine te niz od točno četiri liste od  $n$  brojeva između 1 i 100 (uključivo). Memorija za svaku od te četiri liste i pripadne elemente se pretpostavlja alociranom te inicijaliziranom van funkcije (naravno, svaka lista sadrži brojeve na kartama jednog od igrača). Potrebno je vratiti sortiranu listu koja će predstavljati ispravno rješenje dane razine. Primjerice, ako su u trenutnoj igri na razini 3 dane liste (17,3,87), (4,45,90), (53,100,14) te (22,99,1), ispravno rješenje je lista (1,3,4,14,17,22,45,53,87,90,99,100).

**Napomena:** Nije dozvoljena alokacija, inicijaliziranje ni korištenje dodatnih nizova, struktura i listi te elemenata listi. Dozvoljeno je mijenjanje početnih listi preko pokazivača koji funkcija `rijesi` prima.

## Programiranje 2 – druga popravna provjera znanja, 18. 9. 2020.

**Zadatak 3.** (30 = 20 + 10 bodova) U stringu `p` dan je popis potrepština za trgovinu odvojen znakom `;` pri čemu je prvo zapisan broj potrebnih artikala i odmah nakon njega ime artikla. Npr. `p="10jaje;2mlijeko;1kruh;2maslac;"`. Ime artikla nikad ne počinje brojem i ne sadrži znak `;`.

- (a) Definirajte strukturu `potreba` koja sadrži string `ime` duljine najviše 20 znakova te cijeli broj `kolicina`. Struktura predstavlja artikl na popisu potrepština i njegovu potrebnu količinu. Napišite funkciju `izvadak` koja prima string `p` te vraća niz tipa `potreba` u kojem su zapisani svi artikli iz stringa `p` te njihove količine, te preko varijabilnog argumenta vraća duljinu tog niza.
- (b) Definirajte strukturu `artikl` koja sadrži string `ime` duljine najviše 20 znakova te realni broj `cijena`. Struktura predstavlja ime artikla i njegovu cijenu. Napišite funkciju `iznosRacuna` koji prima string `p`, niz `sviArtikli` tipa `artikl` te cijeli broj `n`. Broj `n` je duljina niza `sviArtikli`. Funkcija `iznosRacuna` treba vratiti iznos računa ako se kupe svi artikli iz niza `p`. Možete pretpostaviti da se svaki artikl u stringu `p` nalazi u nizu `sviArtikli`.

## Programiranje 2 – druga popravna provjera znanja, 18. 9. 2020.

**Zadatak 4.** (25 bodova) U ovom zadatku, riječi su nizovi znakova bez bjelina (*whitespace*). Riječ sadrži barem jedan a najviše 99 znakova. Riječi su međusobno odvojene bjelinama.

U ulaznoj datoteci `rijeci.txt` nalaze se neke riječi. Nakon riječi `AKO` uvijek slijedi riječ koja je dekadski zapis nekog pozitivnog cijelog broja  $n$ . Nakon tog broja slijedi neki niz riječi bez riječi `AKO`, te potom riječ `AKO_KRAJ`. Upravo opisan niz riječi oblika `AKO n ... AKO_KRAJ` nazvat ćemo *blokom*.

Napišite funkciju `void ispis(char *ime)` koja će u datoteku imena koje je argument funkcije ispisati riječi iz datoteke `rijeci.txt`. Na mjestima gdje se nalazi blok treba ispisati riječi koje se nalaze unutar tog bloka ako i samo ako je broj zapisan nakon riječi `AKO` paran. U suprotnom (tj. ako je taj broj neparan) se ništa ne ispisuje na mjestu tog bloka.

Prilikom ispisa u izlaznu datoteku ne morate očuvati točne bjeline iz ulazne datoteke, npr. riječi u izlaznoj datoteci mogu biti svaka u zasebnom retku, bez drugih bjelina.

Možete koristiti pomoćne `char` nizove čiji su broj i veličina poznati prije kompilacije (npr. tri niza po stotinu elemenata).

Primjer ulazne i izlazne datoteke nakon poziva `ispis("izlaz.txt")`:

<code>rijeci.txt</code> :	<code>izlaz.txt</code> (jedan moguć izbor postavljanja bjelina):
AB CD AKO	AB
5 EF AKO_KRAJ AKO 6 GH	CD
	GH
IJ	IJ
AKO_KRAJ AKO_KRA AK 7 A KO 8	AKO_KRA
AKO 4 9 AKO_KRAJ	AK
	7
	A
	KO
	8
	9

Napomene:

- Riječ `AKO_KRAJ` nikad se ne javlja bez pripadne riječi `AKO`.
- Možete koristiti `string.h` i sva druga zaglavlja standardne biblioteke.
- Ne trebate provjeravati uspješnost funkcija za rad s datotekama.