

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Rezultati i uvidi u kolokvije: Rezultati u petak, 3.5., navečer na webu, a uvidi u ponedjeljak, 6.5., u 16 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. (20 bodova) Gusari su stigli s dva broda do otoka na kojem su sakrili blago. Svaka škrinja s blagom je numerirana brojem od 1 do n , $n \leq 20$. Gusari znaju da brodovi imaju nosivost k_1 i k_2 tona, a i -ta škrinja ima masu m_i tona. Međutim, drugi brod ima pregrade u prostoru za teret, tako da u njega možemo utovariti maksimalno $n/4$ škrinja.

- Napišite rekurzivnu funkciju koja će ispisati sve moguće načine za utovarivanje škrinja u ta dva broda, s time da sve škrinje moraju biti utovarene. Dozvoljeno je da jedan od brodova bude prazan. Bitno je samo koja škrinja ide u koji brod, a poredak utovara u pojedini brod nije bitan.
- Svaki način utovara treba ispisati u svom redu, kao uređeni par od dva niza. Prvi niz su indeksi škrinja koje idu u prvi brod i analogno za drugi niz. Ako je neki niz prazan, ne treba ispisati ništa na tom mjestu. Ako nema načina za utovar, ne treba ispisati ništa.

Primjer: Za nosivosti $k_1 = 5$, $k_2 = 3$, i 4 škrinje s masama $m_1 = 1.0$, $m_2 = 1.5$, $m_3 = 2.0$, $m_4 = 2.5$, svi mogući načini utovara su: (1 2 3, 4), (1 2 4, 3).

Napomene: Smijete koristiti jedno polje realnih brojeva koje sadrži mase škrinja i jedno polje cijelih brojeva za spremanje informacija o škrinjama kod utovara. Ne smijete koristiti globalne varijable. Smijete koristiti pomoćne funkcije.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 2. (20 bodova) Ekran nekog računala je matrica znakova `char ekran[10][17]`, visine 10, a širine 17 znakova. Na ekranu može biti iscrtano više pravokutnih prozora. Svaki prozor iscrtan je pomoću nekog svog znaka, različitog od `'.'` i od znaka za ostale prozore. Prozori se nikad ne preklapaju. Lokacije na kojima nema prozora sadrže znak `'.'`. Potrebno je implementirati sljedeće funkcije:

- (a) (4 boda) `int moze(char ekran[10][17], int pi, int pj, int h, int w)` koja provjerava može li se na ekran dodati novi prozor, čiji **gornji lijevi** kut stoji u `ekran[pi][pj]`, a ima visinu `h` i širinu `w`, gdje je `h, w > 0`. Prozor se može dodati ukoliko je u potpunosti unutar ekrana te ukoliko se ne preklapa s drugim prozorima.
- (b) (8 bodova) `void maksimum(char ekran[10][17], int pi, int pj, int *h, int *w)` koja preko varijabilnih parametara vraća dimenzije prozora najveće površine koji se može zauzeti, tako da mu je **gornji lijevi** kut u `ekran[pi][pj]`. Ukoliko je dana lokacija već zauzeta, u `h` i `w` sprema `-1`.
- (c) (8 bodova) `void popuni(char ekran[10][17], char znak)` koja na ekranu pronalazi pravokutni prostor najveće površine koji još nema prozor te na njega crta novi prozor pomoću znaka `znak`.

Na primjer, prva slika prikazuje ekran s dva prozora. Poziv funkcije `moze(ekran, 0, 0, 2, 2)` bi vratio 1, dok bi poziv `moze(ekran, 0, 0, 3, 2)` vratio 0, zbog preklapanja. Poziv funkcije `maksimum(ekran, 7, 0, &k, &l)` bi u `k` spremio 3, a u `l` bi spremio 8. Konačno, poziv `popuni(ekran, '1')` bi rezultirao drugom slikom.

```

.....1111111111
.....1111111111
.xxxxxx.....1111111111
.xxxxxx.....1111111111
.xxxxxx.....1111111111
.xxxxxx.....1111111111
.xxxxxx.....1111111111
.xxxxxx.bbxxxx...
.....bbxxxx...
.....bbxxxx...
.....

```

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 3. (20 bodova) Napišite funkciju `void sort(double ***niznizova, int m, int n)` koja prima niz od m dvodimenzionalnih nizova realnih brojeva (svaki od nizova je veličine $n \times n$) i sortira taj niz nizova uzlazno po zbroju svih elemenata dvodimenzionalnog niza. Primjerice, jedan takav niz nizova, gdje je $m = 4$ i $n = 2$,

$$\begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1.5 & 0 \end{bmatrix}$$

potrebno je sortirati ovako:

$$\begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1.5 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}.$$

Napomena: Sve dodatne nizove koje koristite **dinamički alocirajte** pomoću funkcije `malloc` (ne trebate provjeravati uspješnost alokacije) i na kraju ih **eksplicitno dealocirajte!**

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 4. (15 bodova) Napišite funkciju `zagradi` koja prima string `izraz` u kojem je zapisana riječ. Funkcija treba provjeriti da li se, u svakom početnom komadu te riječi, znak `'a'` pojavljuje više ili jednako puta nego znak `'b'`, pri čemu u cijeloj riječi treba biti jednak broj znakova `'a'` i `'b'`. Ako to vrijedi, funkcija treba u stringu zamijeniti svaki znak `'a'` znakom `'('`, te svaki znak `'b'` znakom `)'`. U protivnom, funkcija ne mijenja string. Kao vrijednost, funkcija treba vratiti broj promijenjenih znakova, te kroz varijabilni argument pokazivač na promijenjeni znak najvećeg indeksa, ako takav postoji, inače `NULL`. Ako je string duljine 0, funkcija treba vratiti 0 kao vrijednost, te `NULL` kroz varijabilni argument. Napišite primjer poziva funkcije. Primjer: String `"-aba-abb"` zadovoljava traženi uvjet, a string `"-aba-bba"` ne zadovoljava.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Rezultati i uvidi u kolokvije: Rezultati u petak, 3.5., navečer na webu, a uvidi u ponedjeljak, 6.5., u 16 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. (20 bodova) Gusari su stigli s dva broda do otoka na kojem su sakrili blago. Svaka škrinja s blagom je numerirana brojem od 1 do n , $n \leq 20$. Gusari znaju da brodovi imaju nosivost k_1 i k_2 tona, a i -ta škrinja ima masu m_i tona. Međutim, u drugi brod polako prodire voda, tako da u njega možemo utovariti samo škrinje mase $m_i \leq k_2/5$ tona.

- Napišite rekurzivnu funkciju koja će ispisati sve moguće načine za utovarivanje škrinja u ta dva broda, s time da sve škrinje moraju biti utovarene. Dozvoljeno je da jedan od brodova bude prazan. Bitno je samo koja škrinja ide u koji brod, a poredak utovara u pojedini brod nije bitan.
- Svaki način utovara treba ispisati u svom redu, kao uređeni par od dva niza. Prvi niz su indeksi škrinja koje idu u prvi brod i analogno za drugi niz. Ako je neki niz prazan, ne treba ispisati ništa na tom mjestu. Ako nema načina za utovar, ne treba ispisati ništa.

Primjer: Za nosivosti $k_1 = 4$, $k_2 = 5$, i 4 škrinje s masama $m_1 = 0.5$, $m_2 = 1.0$, $m_3 = 1.5$, $m_4 = 2.0$, svi mogući načini utovara su: (1 3 4, 2), (3 4, 1 2).

Napomene: Smijete koristiti jedno polje realnih brojeva koje sadrži mase škrinja i jedno polje cijelih brojeva za spremanje informacija o škrinjama kod utovara. Ne smijete koristiti globalne varijable. Smijete koristiti pomoćne funkcije.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 2. (20 bodova) Neko poljoprivredno polje reprezentirano je matricom znakova `char polje[7][13]`. Polje je podijeljeno na pravokutna zemljišta koja pripadaju različitim vlasnicima. Svako zemljište označeno je nekim svojim znakom, različitim od '=' i od znaka za ostala zemljišta. Zemljišta se nikad neće preklapati. Dijelovi polja koji ne pripadaju nikome označeni su znakom '='. Potrebno je implementirati sljedeće funkcije:

- (a) (4 boda) `int isprobaj(char polje[7][13], int pi, int pj, int h, int w)` koja provjerava može li se na polje dodati novo zemljište, čiji **donji desni** kut stoji u `polje[pi][pj]`, a ima visinu `h` i širinu `w`, gdje je `h, w > 0`. Zemljište se može dodati ukoliko je u potpunosti unutar polja te ukoliko se ne preklapa s drugim zemljištima.
- (b) (8 bodova) `void najvece(char polje[7][13], int pi, int pj, int *h, int *w)` koja preko varijabilnih parametara vraća dimenzije zemljišta najveće površine koje se može zauzeti, tako da mu je **donji desni** kut u `polje[pi][pj]`. Ukoliko je dana lokacija već zauzeta, u `h` i `w` sprema `-1`.
- (c) (8 bodova) `void kupi(char polje[7][13], char znak)` koja na polju pronalazi pravokutni prostor najveće površine koji još nema zemljište te na njega dodaje novo zemljište označeno znakom `znak`.

Na primjer, prva slika prikazuje polje s tri zemljišta. Poziv funkcije `isprobaj(polje, 4, 6, 2, 2)` bi vratio `1`, dok bi poziv `isprobaj(polje, 4, 6, 2, 3)` vratio `0`, zbog preklapanja. Poziv funkcije `najvece(polje, 6, 2, &k, &l)` bi u `k` spremio `5`, a u `l` bi spremio `3`. Konačno, poziv `kupi(polje, '1')` bi rezultirao drugom slikom.

```

$$$$=====      $$$$=11111111
$$$$=====      $$$$=11111111
===%=====      ===%11111111
===%=====      ===%=====
===%=====      ===%=====
===%=====      ===%=====
===%=====      ===%=====
=====

```

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 3. (20 bodova) Napišite funkciju `void sort(double ***niznizova, int m, int n)` koja prima niz od m dvodimenzionalnih nizova realnih brojeva (svaki od nizova je veličine $n \times n$) i sortira taj niz nizova silazno po zbroju svih elemenata dvodimenzionalnog niza. Primjerice, jedan takav niz nizova, gdje je $m = 4$ i $n = 2$,

$$\begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1.5 & 0 \end{bmatrix}$$

potrebno je sortirati ovako:

$$\begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1.5 & 0 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}.$$

Napomena: Sve dodatne nizove koje koristite **dinamički alocirajte** pomoću funkcije `malloc` (ne trebate provjeravati uspješnost alokacije) i na kraju ih **eksplicitno dealocirajte!**

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 4. (15 bodova) Napišite funkciju `pregradi` koja prima string `zapis` u kojem je zapisana riječ. Funkcija treba provjeriti da li se, u svakom zadnjem komadu te riječi, znak `'c'` pojavljuje manje ili jednako puta nego znak `'d'`, pri čemu u cijeloj riječi treba biti jednak broj znakova `'c'` i `'d'`. Ako to vrijedi, funkcija treba u stringu zamijeniti svaki znak `'c'` znakom `'{'`, te svaki znak `'d'` znakom `'}'`. U protivnom, funkcija ne mijenja string. Kao vrijednost, funkcija treba vratiti broj promijenjenih znakova, te kroz varijabilni argument pokazivač na promijenjeni znak najmanjeg indeksa, ako takav postoji, inače `NULL`. Ako je string duljine 0, funkcija treba vratiti 0 kao vrijednost, te `NULL` kroz varijabilni argument. Napišite primjer poziva funkcije. Primjer: String `"cdc-cdd-`" zadovoljava traženi uvjet, a string `"dcc-cdd-`" ne zadovoljava.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Rezultati i uvidi u kolokvije: Rezultati u petak, 3.5., navečer na webu, a uvidi u ponedjeljak, 6.5., u 16 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. (20 bodova) Radnici ukrcavaju teret u dva kamiona. Svaka kutija s teretom je numerirana brojem od 1 do n , $n \leq 20$. Radnici znaju da kamioni imaju nosivost k_1 i k_2 kilograma, a i -ta kutija ima masu m_i kilograma. Međutim, drugi kamion ima nosače za ceradu u prostoru za teret, tako da u njega možemo utovariti maksimalno $n/3$ kutija.

- Napišite rekurzivnu funkciju koja će ispisati sve moguće načine za utovarivanje kutija u ta dva kamiona, s time da sve kutije moraju biti utovarene. Dozvoljeno je da jedan od kamiona bude prazan. Bitno je samo koja kutija ide u koji kamion, a poredak utovara u pojedini kamion nije bitan.
- Svaki način utovara treba ispisati u svom redu, kao uređeni par od dva niza. Prvi niz su indeksi kutija koje idu u prvi kamion i analogno za drugi niz. Ako je neki niz prazan, ne treba ispisati ništa na tom mjestu. Ako nema načina za utovar, ne treba ispisati ništa.

Primjer: Za nosivosti $k_1 = 5000$, $k_2 = 3000$, i 4 kutije s masama $m_1 = 1000$, $m_2 = 1500$, $m_3 = 2000$, $m_4 = 2500$, svi mogući načini utovara su: (1 2 3, 4), (1 2 4, 3).

Napomene: Smijete koristiti jedno polje realnih brojeva koje sadrži mase kutija i jedno polje cijelih brojeva za spremanje informacija o kutijama kod utovara. Ne smijete koristiti globalne varijable. Smijete koristiti pomoćne funkcije.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 2. (20 bodova) Neko skladište reprezentirano je matricom znakova `char sklad[8][12]`. Skladište je podijeljeno na pravokutne zone u kojima se čuvaju razni artikli. Svaka zona označena je nekim svojim znakom, različitim od `'`, `'` i od znaka za ostale zone. Zone se nikad ne preklapaju. Lokacije koje ne pripadaju niti jednoj zoni označene su znakom `'`, `'`. Potrebno je implementirati sljedeće funkcije:

- (a) (4 boda) `int moze(char sklad[8][12], int pi, int pj, int h, int w)` koja provjerava može li se u skladište dodati nova zona, čiji **donji lijevi** kut stoji u `sklad[pi][pj]`, a ima visinu `h` i širinu `w`, gdje je `h, w > 0`. Zona se može dodati ukoliko je u potpunosti unutar skladišta te ukoliko se ne preklapa s drugim zonama.
- (b) (8 bodova) `void najveca(char sklad[8][12], int pi, int pj, int *h, int *w)` koja preko varijabilnih parametara vraća dimenzije zone najveće površine koja se može zauzeti, tako da joj je **donji lijevi** kut u `sklad[pi][pj]`. Ukoliko je dana lokacija već zauzeta, u `h` i `w` sprema `-1`.
- (c) (8 bodova) `void zauzmi(char sklad[8][12], char znak)` koja u skladištu pronalazi pravokutni prostor najveće površine na kojem nema drugih zona te na njega dodaje novu zonu označenu znakom `znak`.

Na primjer, prva slika prikazuje skladište s tri zone. Poziv funkcije `moze(sklad, 3, 1, 2, 3)` bi vratio `1`, dok bi poziv `moze(sklad, 3, 1, 3, 3)` vratio `0`, zbog preklapanja. Poziv funkcije `najveca(sklad, 3, 0, &k, &l)` bi u `k` spremio `4`, a u `l` bi spremio `3`. Konačno, poziv `zauzmi(sklad, '1')` bi rezultirao drugom slikom.

```

,,,$$$$,,,,      ,,,$$$$11111
,,,$$$$,,,,      ,,,$$$$11111
,,,%,,,,         ,,,%11111
,,,%,,,,         ,,,%11111
*****,,,,,     *****11111
*****,,,,,     *****11111
*****,,,,,     *****11111
,,,,,,         ,,.,.,11111

```

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 3. (20 bodova) Napišite funkciju `void sort(double ***niznizova, int m, int n)` koja prima niz od m dvodimenzionalnih nizova realnih brojeva (svaki od nizova je veličine $n \times n$) i sortira taj niz nizova uzlazno po umnošku svih elemenata dvodimenzionalnog niza. Primjerice, jedan takav niz nizova, gdje je $m = 4$ i $n = 2$,

$$\begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1.5 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}$$

potrebno je sortirati ovako:

$$\begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1.5 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}.$$

Napomena: Sve dodatne nizove koje koristite **dinamički alocirajte** pomoću funkcije `malloc` (ne trebate provjeravati uspješnost alokacije) i na kraju ih **eksplicitno dealocirajte!**

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 4. (15 bodova) Napišite funkciju `ogradi` koja prima string `opis` u kojem je zapisana riječ. Funkcija treba provjeriti da li se, u svakom početnom komadu te riječi, znak `'e'` pojavljuje više ili jednako puta nego znak `'f'`, pri čemu u cijeloj riječi treba biti jednak broj znakova `'e'` i `'f'`. Ako to vrijedi, funkcija treba u stringu zamijeniti svaki znak `'e'` znakom `'['`, te svaki znak `'f'` znakom `']'`. U protivnom, funkcija ne mijenja string. Kao vrijednost, funkcija treba vratiti broj promijenjenih znakova, te kroz varijabilni argument pokazivač na promijenjeni znak najvećeg indeksa, ako takav postoji, inače `NULL`. Ako je string duljine 0, funkcija treba vratiti 0 kao vrijednost, te `NULL` kroz varijabilni argument. Napišite primjer poziva funkcije. Primjer: String `"-efe-eff"` zadovoljava traženi uvjet, a string `"-efe-ffe"` ne zadovoljava.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Rezultati i uvidi u kolokvije: Rezultati u petak, 3.5., navečer na webu, a uvidi u ponedjeljak, 6.5., u 16 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. (20 bodova) Radnici ukrcavaju teret u dva kamiona. Svaka kutija s teretom je numerirana brojem od 1 do n , $n \leq 20$. Radnici znaju da kamioni imaju nosivost k_1 i k_2 kilograma, a i -ta kutija ima masu m_i kilograma. Međutim, drugom kamionu je probušena guma, tako da u njega možemo utovariti samo kutije mase $m_i \leq k_2/4$ kilograma.

- Napišite rekurzivnu funkciju koja će ispisati sve moguće načine za utovarivanje kutija u ta dva kamiona, s time da sve kutije moraju biti utovarene. Dozvoljeno je da jedan od kamiona bude prazan. Bitno je samo koja kutija ide u koji kamion, a poredak utovara u pojedini kamion nije bitan.
- Svaki način utovara treba ispisati u svom redu, kao uređeni par od dva niza. Prvi niz su indeksi kutija koje idu u prvi kamion i analogno za drugi niz. Ako je neki niz prazan, ne treba ispisati ništa na tom mjestu. Ako nema načina za utovar, ne treba ispisati ništa.

Primjer: Za nosivosti $k_1 = 4000$, $k_2 = 5000$, i 4 kutije s masama $m_1 = 500$, $m_2 = 1000$, $m_3 = 1500$, $m_4 = 2000$, svi mogući načini utovara su: (1 3 4, 2), (3 4, 1 2).

Napomene: Smijete koristiti jedno polje realnih brojeva koje sadrži mase kutija i jedno polje cijelih brojeva za spremanje informacija o kutijama kod utovara. Ne smijete koristiti globalne varijable. Smijete koristiti pomoćne funkcije.

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 2. (20 bodova) Plan nekog grada reprezentiran je matricom znakova `char plan[10][13]`. Plan grada prikazuje zgrade kao pravokutna polja. Svaka zgrada označena je nekim svojim znakom, različitim od '-' i od znaka za ostale zgrade. Zgrade se nikad ne preklapaju. Lokacije na planu koje ne pripadaju niti jednoj zgradi označene su znakom '-'. Potrebno je implementirati sljedeće funkcije:

- (a) (4 boda) `int probaj(char plan[10][13], int pi, int pj, int h, int w)` koja provjerava može li se u plan dodati nova zgrada, čiji **gornji desni** kut stoji u `plan[pi][pj]`, a na planu ima visinu `h` i širinu `w`, gdje je `h, w > 0`. Zgrada se može dodati ukoliko je u potpunosti unutar plana te ukoliko se ne preklapa s drugim zgradama.
- (b) (8 bodova) `void velicina(char plan[10][13], int pi, int pj, int *h, int *w)` koja preko varijabilnih parametara vraća dimenzije zgrade najveće površine na planu koja se može zauzeti, tako da joj je **gornji desni** kut u `plan[pi][pj]`. Ukoliko je dana lokacija već zauzeta, u `h` i `w` sprema `-1`.
- (c) (8 bodova) `void sagradi(char plan[10][13], char znak)` koja u planu pronalazi pravokutni prostor najveće površine na kojem nema drugih zgrada te na njega dodaje novu zgradu označenu znakom `znak`.

Na primjer, prva slika prikazuje plan s tri zgrade. Poziv funkcije `probaj(plan, 3, 3, 2, 4)` bi vratio `1`, dok bi poziv `probaj(plan, 3, 3, 3, 4)` vratio `0`, zbog preklapanja. Poziv funkcije `velicina(plan, 3, 1, &k, &l)` bi u `k` spremio `7`, a u `l` bi spremio `2`. Konačno, poziv `sagradi(plan, '1')` bi rezultirao drugom slikom.

```

-----      ----111111111
$$$$-----  $$$$111111111
$$$$-----  $$$$111111111
-----      ----111111111
-----      ----111111111
---%--*****  ---%--*****
---%--*****  ---%--*****
---%--*****  ---%--*****
---%--*****  ---%--*****
-----      -----
-----      -----

```

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 3. (20 bodova) Napišite funkciju `void sort(double ***niznizova, int m, int n)` koja prima niz od m dvodimenzionalnih nizova realnih brojeva (svaki od nizova je veličine $n \times n$) i sortira taj niz nizova silazno po umnošku svih elemenata dvodimenzionalnog niza. Primjerice, jedan takav niz nizova, gdje je $m = 4$ i $n = 2$,

$$\begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1.5 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}$$

potrebno je sortirati ovako:

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1.5 & 1 \end{bmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}.$$

Napomena: Sve dodatne nizove koje koristite **dinamički alocirajte** pomoću funkcije `malloc` (ne trebate provjeravati uspješnost alokacije) i na kraju ih **eksplicitno dealocirajte!**

Programiranje 2 – prvi kolokvij, 26. 4. 2019.

Zadatak 4. (15 bodova) Napišite funkciju `izgradi` koja prima string `ispis` u kojem je zapisana riječ. Funkcija treba provjeriti da li se, u svakom zadnjem komadu te riječi, znak `'g'` pojavljuje manje ili jednako puta nego znak `'h'`, pri čemu u cijeloj riječi treba biti jednak broj znakova `'g'` i `'h'`. Ako to vrijedi, funkcija treba u stringu zamijeniti svaki znak `'g'` znakom `'<'`, te svaki znak `'h'` znakom `'>'`. U protivnom, funkcija ne mijenja string. Kao vrijednost, funkcija treba vratiti broj promijenjenih znakova, te kroz varijabilni argument pokazivač na promijenjeni znak najmanjeg indeksa, ako takav postoji, inače `NULL`. Ako je string duljine 0, funkcija treba vratiti 0 kao vrijednost, te `NULL` kroz varijabilni argument. Napišite primjer poziva funkcije. Primjer: String `"ghg-ghh-"` zadovoljava traženi uvjet, a string `"hgg-ghh-"` ne zadovoljava.