

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

**Rezultati i uvidi:** Rezultati do petka 13.5.2022. 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!** 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. Uvjete navedene u zadacima (nenegativnost, ograde na  $n$  i sl.) ne treba provjeravati.

**Zadatak 1** (15 bodova) Neka je  $n$  prirodan broj. *Particija* broja  $n$  je prikaz broja  $n$  kao zbroja nekoliko prirodnih brojeva, pri čemu redosljed pribrojnika nije bitan. Na primjer, broj 4 ima 5 particija:  $1 + 1 + 1 + 1$ ,  $1 + 1 + 2$ ,  $1 + 3$ ,  $2 + 2$ ,  $4$ . Može se dokazati da je broj particija od  $n$  u kojima su svi pribrojnici **neparni** jednak broju particija od  $n$  u kojima su svi pribrojnici **različiti**. Na primjer, ako je  $n = 4$ , tada imamo 2 particije na neparne pribrojnike ( $1 + 1 + 1 + 1$ ,  $1 + 3$ ) te 2 particije na različite pribrojnike ( $1 + 3$  i  $4$ ). Vaš je zadatak napisati program koji obavlja numeričku provjeru ove tvrdnje.

Napišite rekursivnu funkciju `particije` koja prima nenegativan cijeli broj i pokazivač na funkciju `provjera` (te po potrebi još neke argumente). Funkcija na koju `provjera` pokazuje prima particiju u obliku niza cijelih brojeva te duljinu niza, tj. broj pribrojnika u particiji. Ona vraća 1 ako particija ima određeno svojstvo te 0 inače. Funkcija `particije` treba računati broj particija danog broja sa svojstvom koje provjerava funkcija na koju pokazuje `provjera`.

U glavnom dijelu programa učitajte prirodan broj  $n$  te ispišite broj particija od  $n$  na neparne i različite pribrojnike. Učinite to na način da funkciji `particije` prosljedite broj  $n$  te pokazivače na funkcije koje provjeravaju jesu li pribrojnici u danoj particiji neparni/različiti (te funkcije također trebete napisati).

**Napomena:** Smijete pretpostaviti da vrijedi  $n \leq 100$ . Za pohranjivanje particija unutar funkcije `particije` smijete koristiti pomoćni niz, no on mora biti memorijske klase `static`!

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

### Zadatak 2 (12+3 bodova)

(a) Napišite funkciju `promijeni` koja prima kvadratnu matricu  $A \in \mathbb{R}^{n \times n}$  reda  $n \leq 20$  i transformira je na sljedeći način:

1. U svakom retku provjerava je li apsolutna vrijednost dijagonalnog elementa veća ili jednaka od zbroja apsolutnih vrijednosti preostalih elemenata u retku (bez dijagonalnog). Ako nije, dijagonalni element se mijenja tako da mu se apsolutna vrijednost poveća za apsolutnu vrijednost te sume. Ako je, nema promjene dijagonalnog elementa u tom retku.
2. Elementarnim transformacijama (zamjenom redaka i stupaca) dovodi element s najvećom apsolutnom vrijednosti na mjesto  $(0, 0)$ .

(b) Napišite dijelove koda funkcije `main` kojima se učitava red kvadratne matrice  $n$  s **komandne linije** te pozove funkcija iz (a) zadatka kako bi transformirala učitavanu matricu. Pretpostavljamo da je  $n \leq 20$ ; matricu ne trebate učitavati.

**Primjer.** Neka je  $n = 5$ , te  $A \in \mathbb{R}^{5 \times 5}$  matrica prikazana dolje lijevo. Nakon svih transformacija dobiva se matrica prikazana dolje desno.

$$\begin{array}{ccccc}
 -13 & 1 & 0 & 5 & 2 \\
 2 & 7 & -2 & 6 & 2 \\
 7 & -5 & -2 & 1 & 0 \\
 -1 & 3 & 2 & 14 & -7 \\
 -1 & 5 & -2 & 1 & 0 \\
 \hline
 \text{polazna matrica}
 \end{array}
 \xrightarrow{\text{1. korak}}
 \begin{array}{ccccc}
 -13 & 1 & 0 & 5 & 2 \\
 2 & \mathbf{19} & -2 & 6 & 2 \\
 7 & -5 & \mathbf{-15} & 1 & 0 \\
 -1 & 3 & 2 & \mathbf{14} & -7 \\
 -1 & 5 & -2 & 1 & \mathbf{9}
 \end{array}
 \xrightarrow{\text{2. korak}}
 \begin{array}{ccccc}
 \mathbf{19} & 2 & -2 & 6 & 2 \\
 1 & -13 & 0 & 5 & 2 \\
 -5 & 7 & -15 & 1 & 0 \\
 3 & -1 & 2 & 14 & -7 \\
 5 & -1 & -2 & 1 & 9 \\
 \hline
 \text{transformirana matrica}
 \end{array}$$

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

**Zadatak 3** (5+10+10\* bodova)

(a) Napišite funkciju

```
int najveci(int** M, int n, int k)
```

koja u  $n \times n$  matrici cijelih brojeva  $M$  nalazi  $k \times k$  podmatricu s najvećom sumom elemenata, te vraća tu sumu. Vrijedi pretpostavka  $0 < k \leq n$  koju nije potrebno provjeravati. Uzastopni elementi  $k \times k$  podmatrice moraju biti uzastopni elementi u matrici  $M$  (vidi primjere ispod - označene su podmatrice s najvećom sumom).

(b) Napišite glavni program (tj. funkciju `main`) koji od korisnika prvo učitava prirodne brojeve  $n$  i  $k$ , pri čemu vrijedi  $k \leq n$  (što ne treba provjeravati). Zatim se od korisnika učitava niz matrica čiji su elementi cijeli brojevi (ovdje učitati matricu znači redom učitati sve njezine elemente i spremiti ih redom u tu matricu po retcima). Pritom je svaka matrica dimenzije  $n \times n$ , a učitavanje završava kada je unesena nul-matrica (tj. matrica čiji su svi elementi jednaki nula - ta matrica je također dio tog niza!). Program treba sortirati (zamjenom pokazivača) učitani niz matrica **silazno** po vrijednosti koju za matricu daje funkcija iz a) podzadatka (za  $k$  koji je program učitao na početku). Ukoliko za neke dvije matrice funkcija iz a) podzadatka vraća istu vrijednost, tada nam poredak takvih matrica nije bitan. Funkciju iz a) podzadatka smijete koristiti čak i ako ju niste napisali. Na kraju programa, ispišite dobiveni niz.

Primjer sortiranog niza matrica (za  $n = 4$  i  $k = 2$ ; matrice su unesene u nekom drugom poretku, a za svaku je označena  $k \times k$  podmatrica s najvećom sumom):

$$\begin{pmatrix} 0 & 0 & 0 & 9 \\ 4 & 5 & -2 & 1 \\ \boxed{3} & \boxed{1} & 2 & 5 \\ 4 & 5 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 2 & 1 & 3 \\ \boxed{5} & \boxed{0} & 1 & 2 \\ 3 & 4 & 0 & 4 \end{pmatrix}, \begin{pmatrix} \boxed{4} & \boxed{5} & 2 & 1 \\ 0 & 2 & 1 & 3 \\ 4 & 0 & 1 & 2 \\ 4 & 2 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 3 & 0 & 2 \\ 0 & \boxed{5} & \boxed{0} & 1 \\ 0 & \boxed{2} & \boxed{3} & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{0} & \boxed{0} \\ 0 & 0 & \boxed{0} & \boxed{0} \end{pmatrix}.$$

**Za dodatnih 10\* bodova:** U (b) podzadatku sortirajte niz matrica korištenjem algoritma čija je prosječna vremenska složenost  $\mathcal{O}(n \log n)$ . Dodatni bodovi ne broje se u traženih 80% bodova na jednom programskom zadatku, tj. 80% se na ovome zadatku računa od 15 bodova ( $80\% \cdot 15 = 12$  bodova).

**Važno:** Svi nizovi korišteni u ovome zadatku moraju biti dinamički (re)alocirani (i na kraju dealocirani!), te moraju zauzimati točno onoliko memorije koliko je potrebno! Ne treba provjeravati uspješnost (re)alokacije. Uz standardnu `stdio.h` biblioteku, u ovome zadatku dozvoljeno je korištenje još jedino `stdlib.h` biblioteke.

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

**Zadatak 4** (15 bodova) Napišite funkciju `vlak` koja prima string `s` u kojem je spremljen vozni red vlaka. Vozni red se sastoji od niza vremena oblika `hh:mm` međusobno odvojenih znakom `'-'`. Prvo vrijeme odgovara vremenu polaska s polaznog stajališta, a zadnje vremenu dolaska na krajnje odredište. Nakon što vozač odveze zadanu rutu, odmara se sat vremena, a zatim istom rutom vraća natrag. Funkcija treba promijeniti string `s` tako da se u njemu nalazi vozni red povratka nakon pauze od sat vremena. Dodatno, preko varijabilnog argumenta, funkcija vraća pokazivač na prvo vrijeme nakon `16:00` u promijenjenom stringu ili `NULL` ukoliko takvo vrijeme ne postoji. Možete pretpostaviti da će se povratak dogoditi unutar 24 sata.

**Napomena:** Zabranjeno je korištenje biblioteka `string.h` i `ctype.h`.

**Primjer:** `"09:15-10:00-12:30-13:10"` → `"14:10-14:50-17:20-18:05"`.

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

**Rezultati i uvidi:** Rezultati do petka 13.5.2022. 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!** 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. Uvjete navedene u zadacima (nenegativnost, ograde na  $n$  i sl.) ne treba provjeravati.

**Zadatak 1** (15 bodova) Neka je  $n$  prirodan broj. *Particija* broja  $n$  je prikaz broja  $n$  kao zbroja nekoliko prirodnih brojeva, pri čemu redoslijed pribrojnika nije bitan. Na primjer, broj 4 ima 5 particija:  $1 + 1 + 1 + 1$ ,  $1 + 1 + 2$ ,  $1 + 3$ ,  $2 + 2$ ,  $4$ . Može se dokazati da je zbroj brojeva **jedinica** po svim particijama od  $n$  jednak zbroju brojeva **različitih pribrojnika** po svim particijama od  $n$ . Na primjer, ako je  $n = 4$ , tada navedene particije sadrže redom 4, 2, 1, 0, 0 jedinica te imaju redom 1, 2, 2, 1, 1 različitih pribrojnika, pri čemu vrijedi  $4 + 2 + 1 + 0 + 0 = 7$  i  $1 + 2 + 2 + 1 + 1 = 7$ . Vaš je zadatak napisati program koji obavlja numeričku provjeru ove tvrdnje.

Napišite rekurzivnu funkciju **particije** koja prima nenegativan cijeli broj i pokazivač na funkciju **prebroji** (te po potrebi još neke argumente). Funkcija na koju **prebroji** pokazuje prima particiju u obliku niza cijelih brojeva te duljinu niza, tj. broj pribrojnika u particiji, a vraća nenegativan cijeli broj. Funkcija **particije** treba računati zbroj povratnih vrijednosti funkcije na koju **prebroji** pokazuje po svim particijama od  $n$ .

U glavnom dijelu programa učitajte prirodan broj  $n$  te ispišite zbroj brojeva jedinica i brojeva različitih pribrojnika po svim particijama od  $n$ . Učinite to na način da funkciji **particije** prosljedite broj  $n$  te pokazivače na funkcije koje broje koliko u danoj particiji ima jedinica/različitih pribrojnika (te funkcije također trebate napisati).

**Napomena:** Smijete pretpostaviti da vrijedi  $n \leq 100$ . Za pohranjivanje particija unutar funkcije **particije** smijete koristiti pomoćni niz, no on mora biti memorijske klase **static**!

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

### Zadatak 2 (12+3 bodova)

(a) Napišite funkciju `promijeni` koja prima kvadratnu matricu  $A \in \mathbb{R}^{n \times n}$  reda  $n \leq 30$  i transformira je na sljedeći način:

1. U svakom stupcu provjerava je li apsolutna vrijednost dijagonalnog elementa veća ili jednaka od zbroja apsolutnih vrijednosti preostalih elemenata u stupcu (bez dijagonalnog). Ako nije, dijagonalni element se mijenja tako da mu se apsolutna vrijednost poveća za apsolutnu vrijednost te sume. Ako je, nema promjene dijagonalnog elementa u tom stupcu.
2. Elementarnim transformacijama (zamjenom redaka i stupaca) dovodi element s najvećom apsolutnom vrijednosti na mjesto  $(n-1, n-1)$ .

(b) Napišite dijelove koda funkcije `main` kojima se učita red kvadratne matrice  $n$  s **komandne linije** te pozove funkcija iz (a) zadatka kako bi transformirala učitanu matricu. Pretpostavljamo da je  $n \leq 30$ ; matricu ne trebate učitati.

**Primjer.** Neka je  $n = 5$ , te  $A \in \mathbb{R}^{5 \times 5}$  matrica prikazana dolje lijevo. Nakon svih transformacija dobiva se matrica prikazana dolje desno.

$$\begin{array}{ccccc}
 -13 & 1 & 0 & 5 & 2 \\
 2 & 7 & -2 & 6 & 2 \\
 3 & -5 & -2 & 1 & 0 \\
 -1 & 3 & 2 & 14 & -7 \\
 -1 & 5 & -2 & 1 & 0 \\
 \hline
 \text{polazna matrica}
 \end{array}
 \xrightarrow{\text{1. korak}}
 \begin{array}{ccccc}
 -13 & 1 & 0 & 5 & 2 \\
 2 & \mathbf{21} & -2 & 6 & 2 \\
 3 & -5 & \mathbf{-8} & 1 & 0 \\
 -1 & 3 & 2 & \mathbf{14} & -7 \\
 -1 & 5 & -2 & 1 & \mathbf{11}
 \end{array}
 \xrightarrow{\text{2. korak}}
 \begin{array}{ccccc}
 -13 & 2 & 0 & 5 & 1 \\
 -1 & 11 & -2 & 1 & 5 \\
 3 & 0 & -8 & 1 & -5 \\
 -1 & -7 & 2 & 14 & 3 \\
 2 & 2 & -2 & 6 & \mathbf{21} \\
 \hline
 \text{transformirana matrica}
 \end{array}$$

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

**Zadatak 3** (5 + 10 + 10\* bodova)

(a) Napišite funkciju

```
double najmanji(double** M, int n, int m)
```

koja u  $n \times n$  matrici realnih brojeva  $M$  nalazi  $m \times m$  podmatricu s najmanjim umnoškom elemenata, te vraća taj umnožak. Vrijedi pretpostavka  $0 < m \leq n$  koju nije potrebno provjeravati. Uzastopni elementi  $m \times m$  podmatrice moraju biti uzastopni elementi u matrici  $M$  (vidi primjere ispod - označene su podmatrice s najmanjim umnoškom).

(b) Napišite glavni program (tj. funkciju `main`) koji od korisnika prvo učitava prirodne brojeve  $n$  i  $m$ , pri čemu vrijedi  $m \leq n$  (što ne treba provjeravati). Zatim se od korisnika učitava niz matrica čiji su elementi realni brojevi (ovdje učitati matricu znači redom učitati sve njezine elemente i spremiti ih redom u tu matricu po retcima). Pritom je svaka matrica dimenzije  $n \times n$ , a učitavanje završava kada je unesena matrica čiji su svi elementi jednaki 1 (ona je također dio tog niza!). Program treba sortirati (zamjenom pokazivača) učitani niz matrica **silazno** po vrijednosti koju za matricu daje funkcija iz a) podzadatka (za  $m$  koji je program učitao na početku). Ukoliko za neke dvije matrice funkcija iz a) podzadatka vraća istu vrijednost, tada nam poredak takvih matrica nije bitan. Funkciju iz a) podzadatka smijete koristiti čak i ako ju niste napisali. Na kraju programa, ispišite dobiveni niz.

Primjer sortiranog niza matrica (za  $n = 4$  i  $m = 2$ ; matrice su unesene u nekom drugom poretku, a za svaku je označena  $m \times m$  podmatrica s najmanjim umnoškom):

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & \boxed{1} & \boxed{1} \\ 1 & 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 3.2 & 0.1 \\ -0.3 & 3 & 4.2 & -4 \\ \boxed{2} & \boxed{3.4} & 2.1 & -0.5 \\ -0.6 & 3 & 2 & -1 \end{pmatrix}, \begin{pmatrix} 1.1 & 2.4 & 3 & 1 \\ 2.3 & \boxed{4} & \boxed{5.1} & 2 \\ -1 & \boxed{3} & \boxed{-2.1} & 3.2 \\ 3.1 & 5.2 & 1 & 0 \end{pmatrix}.$$

**Za dodatnih 10\* bodova:** U (b) podzadatku sortirajte niz matrica korištenjem algoritma čija je prosječna vremenska složenost  $\mathcal{O}(n \log n)$ . Dodatni bodovi ne broje se u traženih 80% bodova na jednom programskom zadatku, tj. 80% se na ovome zadatku računa od 15 bodova ( $80\% \cdot 15 = 12$  bodova).

**Važno:** Svi nizovi korišteni u ovome zadatku moraju biti dinamički (re)alocirani (i na kraju dealocirani!), te moraju zauzimati točno onoliko memorije koliko je potrebno! Ne treba provjeravati uspješnost (re)alokacije. Uz standardnu `stdio.h` biblioteku, u ovome zadatku dozvoljeno je korištenje još jedino `stdlib.h` biblioteke.

## Programiranje 2 – prvi kolokvij, 6. 5. 2022.

**Zadatak 4** (15 bodova) Napišite funkciju `autobus` koja prima string `s` u kojem je spremljen vozni red autobusa. Vozni red se sastoji od niza vremena oblika `hh:mm` međusobno odvojenih znakom `*`. Prvo vrijeme odgovara vremenu polaska s polaznog stajališta, a zadnje vremenu dolaska na krajnje odredište. Nakon što vozač odveze zadanu rutu, odmara se sat vremena, a zatim istom rutom vraća natrag. Funkcija treba promijeniti string `s` tako da se u njemu nalazi vozni red povratka nakon pauze od sat vremena. Dodatno, preko varijabilnog argumenta, funkcija vraća pokazivač na prvo vrijeme nakon 14:00 u promijenjenom stringu ili `NULL` ukoliko takvo vrijeme ne postoji. Možete pretpostaviti da će se povratak dogoditi unutar 24 sata.

**Napomena:** Zabranjeno je korištenje biblioteka `string.h` i `ctype.h`.

**Primjer:** "08:15\*09:00\*11:30\*12:10" → "13:10\*13:50\*16:20\*17:05".