

Programiranje 1 – 1. ispit, 7. 2. 2025.

Upute: Na ispitus 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. Zadaci 3 – 5 su programski s obzirom na uvjet polaganja.

U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Zadatak 1 ($4+3+5+6=18$ bodova)

- (a) Odredite sve baze $b \geq 2$ i uvjete na varijable x, y , takve da vrijedi $2 \cdot (xy)_b > (yx)_{b+1}$.
- (b) Izračunajte bez pretvaranja u bazu 10: $((JJJ)_{(20)} - (999)_{(20)}) \cdot (21)_{20}$.
- (c) Neka je x 3-bitni prirodni broj. Napravite logički sklop koji računa je li zadani broj x djeljiv s 3. **Obavezno napišite cijeli postupak.**
- (d) Unutar teksta se nalaze imena, prezimena korisnika (počinju velikim slovom) i korisnička imena zadana na sljedeći način: **Ime Prezime - xyBroj**, gdje x predstavlja prva 3 slova imena, y prva tri slova prezimena, a *Broj* je niz od 0 do maksimalno 3 znamenke. Npr. za korisnika Pero Peric, tražena konstrukcija unutar teksta može biti PerPer ili PerPer3 ili PerPer456 (velika slova imena i prezimena ostaju velika i u korisničkom imenu).
 - a) Napišite regularni izraz koji prepoznaje ime, prezime i korisničko ime zadano u gore opisanom formatu u proizvoljnom tekstu.
 - b) Napišite regularni izraz koji prepoznaje niz imena, prezimena i korisničkih imena odvojenih zarezom. Npr. prepoznaje Pero Peric - PerPer, Mirko Miric - MirMir34, Damir Misko - DamMis742

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 2 (3+6+6 = 15 bodova)

- a) Ako želimo spremiti cijeli broj na veličinu jednog byta tipično se koristi zaglavlje `stdint.h` gdje postoji tip `int8_t`. U slučaju da nemate zaglavlje `stdint.h` kako biste kreirali gornji tip imena `INT8_T` koristeći naredbu `typedef`

- b) Inicijaliziramo tri varijable `a, b, c` tipa `INT8_T`

```
INT8_T a=1;
```

```
INT8_T b=4;
```

```
INT8_T c=127;
```

Što vraća `sizeof(c)`?

Napišite redom ispis operacija:

```
printf("%d", a+b);
```

```
printf("%d", a&& b);
```

```
printf("%d", a&b); /* logicko i bit po bit */
```

```
c=c+a;
```

```
printf("%d", c);
```

```
c=a<<1; /* pomak bitova u lijevo */
```

```
printf("%d", c);
```

- c) Napišite funkciju koja prima cjelobrojni tip `INT8_T` i ispisuje prikaz tog broja u računalu kao niz bitova.

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 3 (12+8 = 20 bodova)

Za polinom $p(x) = \sum_{i=0}^n a_i x^i$ definiramo njegovu k -tu derivaciju kao polinom

$$p^{(k)}(x) = \sum_{i=k}^n \frac{i!}{(i-k)!} a_i x^{i-k}.$$

Ako u točki $c \in \mathbb{R}$ polinom p ima ekstrem, onda je $p'(c) = 0$. Više derivacije nam daju dovoljan uvjet za određivanje je li to točka minimuma ili točka maksimuma. Za polinome stupnja barem dva postoji $m \in \mathbb{N}$, takav da za svaki $k \in \{1, \dots, m\}$ vrijedi $p^{(k)}(c) = 0$ i $p^{(m+1)}(c) \neq 0$, te tada u slučaju

- kada je $p^{(m+1)}(x) > 0$, p ima u x strogi lokalni minimum,
- kada je $p^{(m+1)}(x) < 0$, p ima u x strogi lokalni maksimum.

- Napišite funkciju `int ekstrem(double a[], int n, double c)` koja prima cijeli broj $n \geq 2$, niz koeficijenata $a_0, a_1, \dots, a_n$ ($a_n \neq 0$) polinoma p te realan broj c . Funkcija za točku lokalnog ekstrema c polinoma p (ne treba provjeravati) evaluacijom viših derivacija Hornerovim algoritmom treba odrediti vrstu ekstrema te vratiti 0 ako je c strogi lokalni minimum, inače 1.
- Napišite funkciju `int minimumi(double a[], int n, double x[], int l)` koja prima cijeli broj $n \geq 2$, niz koeficijenata $a_0, a_1, \dots, a_n$ polinoma p , te niz točaka ekstrema $x_0, x_1, \dots, x_{l-1}$ polinoma p duljine l . Funkcija treba izbrisati iz niza sve točke maksimuma tako da ostanu samo točke minimuma te vratiti novu duljinu niza. Smijete koristiti funkciju `ekstrem` iz (a) dijela zadatka čak i ako ju niste napisali.

Napomena. Dozvoljeno je korištenje dodatnih (pomoćnih) funkcija. Zabranjeno je korištenje dodatnih nizova.

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 4 (14 + 13 = 27 bodova) Slavko je vlasnik zalagaonice, i ima $n \leq 200$ artikala koji su kodirani sa jedinstvenim znakom i zapisani u niz **char** $A[]$ a njihove cijene su zapisane u nizu **double** $C[]$ na način da je $C[i]$ cijena od artikla $A[i]$.

redni broj	Artikl	Cijena/€
0	a	2.50
1	E	55.99
2	h	12.20
3	Q	11.00

- (a) Kako bi Slavko imao pregledniju zalagaonicu, napišite funkciju koja sortira artikle uzlazno po cijeni koristeći algoritam sortiranja umetanjem (*Insertion sort*). Ukoliko podzadatak točno riješite nekim drugim algoritmom sortiranja, vrednovat će se sa 7 bodova.
- (b) Pod pretpostavkom da je niz artikala sortirani uzlazno po cijeni, napišite funkciju

char Ponuda(**char** Artikl[], **double** Cijena[], **int** n , **double** N)

koja, koristeći binarno pretraživanje, vraća znak najskupljeg artikla koju kupac može priuštiti količinom novca N .

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 5 ($10 + 10 = 20$ bodova)

- a) Napišite funkciju `int najcesca_znamenka(int n, int b, int *broj_pojavljivanja)` koja prima prirodne brojeve n i b ($b \geq 2$, ne treba provjeravati) i vraća onu znamenku koja se u zapisu broja n u bazi b javlja najviše puta. Ako takvih ima više, treba vratiti najveću. Kroz varijabilni argument, funkcija treba vratiti broj pojavljivanja najčešće znamenke.
- b) Napišite funkciju `int nadi_bazu(int n, int k)` koja prima prirodne brojeve n i k te računa najmanju bazu $b \geq 2$ takvu da zapis broja n u bazi b ima k znamenaka. Ako takva ne postoji, neka funkcija vraća -1 .

Programiranje 1 – 1. ispit, 7. 2. 2025.

Upute: Na ispitus 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. Zadaci 3 – 5 su programski s obzirom na uvjet polaganja.

U svim zadacima zabranjeno je korištenje dodatnih nizova i standardne matematičke biblioteke (zaglavlje `math.h`), osim ako je u zadatku drugačije navedeno.

Zadatak 1 ($4+3+5+6=18$ bodova)

- (a) Odredite sve baze $b \geq 2$ i uvjete na varijable x, y , takve da vrijedi $2 \cdot (xy)_b > (yx)_{b+2}$.
- (b) Izračunajte bez pretvaranja u bazu 10: $((III)_{(19)} - (999)_{(19)}) \cdot (23)_{(19)}$.
- (c) Neka je x 3-bitni prirodni broj. Napravite logički sklop koji računa je li zadani broj x složen. **Obavezno napišite cijeli postupak.**
- (d) Unutar teksta se nalaze imena, prezimena korisnika (počinju velikim slovom) i korisnička imena zadana na sljedeći način: **Ime Prezime - xyBroj**, gdje x predstavlja prva 3 slova imena, y prva tri slova prezimena, a *Broj* je niz od 1 do maksimalno 3 znamenke. Npr. za korisnika Pero Peric, tražena konstrukcija unutar teksta može biti PerPer1 ili PerPer32 ili PerPer456 (velika slova imena i prezimena ostaju velika i u korisničkom imenu).
 - a) Napišite regularni izraz koji prepoznaje ime, prezime i korisničko ime zadano u gore opisanom formatu u proizvoljnom tekstu.
 - b) Napišite regularni izraz koji prepoznaje niz imena, prezimena i korisničkih imena odvojenih točkom zarezom. Npr. prepoznaje Pero Peric - PerPer1; Mirko Miric - MirMir34; Damir Misko - DamMis742

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 2 (3+6+6=15 bodova)

- a) Ako želimo spremiti cijeli broj na veličinu jednog byta tipično se koristi zaglavlje `stdint.h` gdje postoji tip `int8_t`. U slučaju da nemate zaglavlje `stdint.h` kako biste kreirali gornji tip imena `INT8_T` koristeći naredbu `typedef`

- b) Inicijaliziramo tri varijable `a, b, c` tipa `INT8_T`

```
INT8_T a=1;
```

```
INT8_T b=4;
```

```
INT8_T c=127;
```

Što vraća `sizeof(c)`?

Napišite redom ispis operacija:

```
printf("%d", a+b);
```

```
printf("%d", a&& b);
```

```
printf("%d", a&b); /* logicko i bit po bit */
```

```
c=c+a;
```

```
printf("%d", c);
```

```
c=a<<1; /* pomak bitova u lijevo */
```

```
printf("%d", c);
```

- c) Napišite funkciju koja prima cjelobrojni tip `INT8_T` i ispisuje prikaz tog broja u računalu kao niz bitova.

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 3 (12+8 = 20 bodova)

Za polinom $q(x) = \sum_{i=0}^m b_i x^i$ definiramo njegovu k -tu derivaciju kao polinom

$$q^{(k)}(x) = \sum_{i=k}^m \frac{i!}{(i-k)!} b_i x^{i-k}.$$

Ako u točki $c \in \mathbb{R}$ polinom q ima ekstrem, onda je $q'(c) = 0$. Više derivacije nam daju dovoljan uvjet za određivanje je li to točka minimuma ili točka maksimuma. Za polinome stupnja barem dva postoji $n \in \mathbb{N}$, takav da za svaki $k \in \{1, \dots, n\}$ vrijedi $q^{(k)}(c) = 0$ i $q^{(n+1)}(c) \neq 0$, te tada u slučaju

- kada je $q^{(n+1)}(c) > 0$, p ima u x strogi lokalni minimum,
- kada je $q^{(n+1)}(c) < 0$, p ima u x strogi lokalni maksimum.

- Napišite funkciju `int ekstrem(double b[], int m, double c)` koja prima cijeli broj $m \geq 2$, niz koeficijenata $b_0, b_1, \dots, b_m$ ($b_m \neq 0$) polinoma q te realan broj c . Funkcija za točku lokalnog ekstrema c polinoma q (ne treba provjeravati) evaluacijom viših derivacija Hornerovim algoritmom treba odrediti vrstu ekstrema te vratiti 0 ako je c strogi lokalni minimum, inače 1.
- Napišite funkciju `int minimumi(double a[], int m, double x[], int l)` koja prima cijeli broj $m \geq 2$, niz koeficijenata $b_0, b_1, \dots, b_m$ polinoma q , te niz točaka ekstrema $x_0, x_1, \dots, x_{l-1}$ polinoma q duljine l . Funkcija treba izbrisati iz niza sve točke maksimuma tako da ostanu samo točke minimuma te vratiti novu duljinu niza. Smijete koristiti funkciju `ekstrem` iz (a) dijela zadatka čak i ako ju niste napisali.

Napomena. Dozvoljeno je korištenje dodatnih (pomoćnih) funkcija. Zabranjeno je korištenje dodatnih nizova.

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 4 (14 + 13 = 27 bodova) Mirko je vlasnik štanda s lubenicama, i ima $n \leq 200$ lubenica koje su kodirani sa jedinstvenim znakom i zapisani u niz **char** $L[]$ a njihove mase su zapisane u nizu **double** $M[]$ na način da je $M[i]$ masa od lubenice $L[i]$.

redni broj	Lubenica	Masa/kg
0	O	20.50
1	f	54.99
2	a	12.20
3	T	90.00

- (a) Kako bi Mirko imao pregledniji štand, napišite funkciju koja sortira lubenice uzlazno po masi koristeći algoritam sortiranja umetanjem (*Insertion sort*). Ukoliko podzadatak točno riješite nekim drugim algoritmom sortiranja, vrednovat će se sa 7 bodova.
- (b) Pod pretpostavkom da je niz lubenica sortiran uzlazno po masi, napišite funkciju

char Ponuda(**char** Lubenica[], **double** Masa[], **int** n , **double** T)

koja, koristeći binarno pretraživanje, vraća znak najteže lubenice koju kupac može nositi ako je maksimalno opterećenje koje kupac može podnijeti T kilograma.

Programiranje 1 – 1. ispit, 7. 2. 2025.

Zadatak 5 ($10 + 10 = 20$ bodova)

- a) Napišite funkciju `int najcesca_znamenka(int n, int b, int *broj_pojavljivanja)` koja prima prirodne brojeve n i b ($b \geq 2$, ne treba provjeravati) i vraća onu znamenku koja se u zapisu broja n u bazi b javlja najviše puta. Ako takvih ima više, treba vratiti najmanju. Kroz varijabilni argument, funkcija treba vratiti broj pojavljivanja najčešće znamenke.
- b) Napišite funkciju `int nadi_bazu(int n, int k)` koja prima prirodne brojeve n i k te računa najmanju bazu $b \geq 2$ takvu da zapis broja n u bazi b ima k znamenaka. Ako takva ne postoji, neka funkcija vraća -1 .