

Rješenja zadataka 1 - 4 napisati na vlastitim papirima i predati ih u košuljici.

Rješenja zadataka 5 - 7 napisati na za to predviđeno mjestu uz tekst zadatka te list sa zadacima predati u košuljici.

1. (18 bodova)

Svaka stavka binarne datoteke osobe.bin sadrži zapis o jednoj osobi, a čine ju: ime i prezime, svako duljine 20+1 karaktera (char), starost osobe izražena u godinama (int), te sifra koja identifikira zanimanje te osobe prema datoteci zanimanja.bin.

U binarnoj datoteci zanimanja.bin nalaze se zapisi koji predstavljaju zanimanja. Svaki zapis te datoteke se sastoji od jedinstvene sifre zanimanja (int) i naziva zanimanja duljine 20+1 karaktera (char). Redni broj zapisa u datoteci zanimanja.bin odgovara sifri zanimanja.

Za svaki zapis iz datoteke osobe.bin pronaći adekvatan zapis iz datoteke zanimanja.bin i u datoteku izlaz.txt zapisati podatke o imenu, prezimenu i zanimanju svake osobe, onim redoslijedom kojim se pojavljuju u datoteci osobe.bin. Na kraj datoteke izlaz.txt potrebno je zapisati i prosječnu starost svih osoba.

Svaku datoteku je dozvoljeno samo jednom otvoriti isključivo za čitanje ili pisanje.

Primjer zapisa osobe.bin

```
("Pero", "Peric", 55, 2);
("Ivo", "Ivic", 45, 3);
("Luj", "Lujic", 22, 1);
("Vilbor", "Ivanic", 31, 3);
("Danko", "Dancic", 75, 4);
```

Primjer zapisa zanimanja.bin

```
(1, "stolar");
(2, "znanstvenik");
(3, "kuhar");
(4, "elektricar");
(5, "knjiznicar");
```

Pero · Peric · znanstvenik
Ivo · Ivic · kuhar
Luj · Lujic · stolar
Vilbor · Ivanic · znanstvenik
Danko · Dancic · elektricar
Prosječna starost: 39

2. (17 bodova)

Kvadratna matrica nenegativnih cjelobrojnih vrijednosti (int) prikazuje brojeve golova koji su ostvareni na susretima rukometnog ligaškog turnira. Polje u matrici (i, j) odgovara broju golova koje je na susretu ostvario i-ti tim igrajući protiv j-tog tima.

Primjer: zaokružena vrijednost 29 predstavlja broj golova koji je 0. tim postigao u susretu s timom 1.

| | | | | |
|----|----|----|----|----|
| 0 | 29 | 12 | 34 | 5 |
| 13 | 0 | 24 | 7 | 3 |
| 4 | 32 | 0 | 5 | 23 |
| 29 | 10 | 17 | 0 | 34 |
| 15 | 23 | 26 | 31 | 0 |

Za pobjedu na susretu tim dobiva 2 boda, za izjednačen rezultat 1 bod, a za izgubljen susret dobiva 0 bodova. Dijagonale matrice su popunjene nulama.

Napisati funkciju brojBodovaTima koja za matricu golova reda N i indeks tima vraća ukupan broj bodova ostvarenih na turniru za tim s danim indeksom.

Napisati funkciju glavnog programa (main) u kojoj se inicijalizira prikazana matrica (pomoću inicijalizatora, ne učitavanjem sa tipkovnice), te pozivom funkcije brojBodovaTima određuje i na zaslon ispisuje indeks tima koji je ostvario najbolji rezultat (u slučaju jednakog broja bodova ispisati indeks tima s manjom vrijednošću).

3. (17 bodova)

Struktura s_točka sadrži podatke o koordinatama jedne točke u dvodimenzionalnom koordinatnom sustavu, a sastoji se od X i Y koordinata (double). Napisati strukturu s_točka.

Napisati funkciju racUdaljenost koja će za zadano jednodimenzijско polje struktura s_točka i duljinu polja n (int) vratiti duljinu (double) puta koji redom prolazi svim točkama polja.

Npr. Ako su redom elementi bolja točke (2,4), (5,0), (0,0) na zaslonu se treba ispisati vrijednost 10.00.

Napomena: formula za izračun udaljenosti dviju točaka u ravni: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

4. (16 bodova)

Implementirati funkciju brojiBajtove koja vraća broj bajtova 64-bitnog cijelog broja koji se podudaraju sa uzorkom broja veličine jednog bajta. Funkcija treba pratiti sljedeći prototip:

```
int brojiBajtove(unsigned long long broj, unsigned const char uzorak)
```

```

broj: 0xA132A3FA132A3FA, usorak: 0xA1; rezultat: 4
broj: 0xAFF7204DFF7204DFF, usorak: 0xFF; rezultat: 2;

```

Napomena: Za implementaciju funkcije nije dozvoljeno korištenje aritmetičkih operacija (osim inkrementa i dekrementa) i sgraničnih tipova podataka.

5. (4 boda)

Prikazati što će se tijekom izvršavanja odjeljka programa ispisati na zaslonu.

```

#include <stdio.h>
#include <string.h>

int main(void) {
    char s1[15] = "student";
    char s2[4] = "ucl";

    printf("%s", strcpy(s1[8], s2));
}

```

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|
| s | t | u | d | e | n | t | u | c | l | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|

6. (4 boda)

Popunite praznine u sljedećem programskom odjeljku kako bi se ostvarilo ispravno generiranje pseudoslužajnog cijelog brojeva iz raspona [1, 30000] i pohrana istog u varijablu rezultat.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

rand((unsigned)time(0));
int rezultat = (int)(rand() % (30000 - 1) + 1);

```

7. (4 boda)

Kaskadnu selekciju prikazanu u sljedećem programskom odjeljku realizirati minimalnim korištenjem grananja (naredbom if-else).

```

int rezultat = 1, vrijednost;
scanf("%d", &vrijednost);

switch (vrijednost) {
    case 0:
        rezultat = 0;
        break;
    case 1:
    case 2:
    case 3:
    case 9:
        rezultat += 7;
        break;
    case 7:
    case 10:
        rezultat += 10;
    default:
        rezultat += 5;
}

```

Prostor za odgovor

int rezultat = 1, vrijednost;

scanf("%d", &vrijednost);

if (vrijednost == 0) rezultat = 0;

else if (vrijednost == 1 || vrijednost == 2 || vrijednost == 3 || vrijednost == 9) {
 rezultat += 7;
}

else if (vrijednost == 7 || vrijednost == 10) rezultat += 10;

else rezultat += 5;