

Are you sure you want to submit the exam?

My answers:

1

Napisati funkciju generirajSlucajniRedak prototipa:

```
void generirajSlucajniRedak(int n, int *mat)
```

koja kao parametre prima pokazivač na prvi element dvodimenzijskog polja cijelih brojeva (kvadratna matrica) i cijeli broj n koji predstavlja broj redaka kvadratne matrice. Funkcija mora prvo slučajno izabrati redak koji će "generirati", a zatim taj redak popuniti slučajno izabranim vrijednostima iz intervala [2, 8].

Inicijalna vrijednost generatora pseudoslučajnih brojeva (pozivom funkcije srand) se postavlja na pozivajućoj razini (npr. u funkciji main) i nemojte ju ponovno postavljati.

Napomena: Interval pseudoslučajnih brojeva prilagoditi traženom intervalu uz pomoć izraza koji koristi operaciju ostatka cjelobrojnog dijeljenja (prikazan na predavanjima).

My answer:

```
void generirajSlucajniRedak(int n, int *mat) {
    int (*matrica)[n] = (void*)mat, i, j, random = rand() % n;

    for(i = 0; i < n; i++) {
        if(i != random) continue;
        for(j = 0; j < n; j++)
            matrica[i][j] = rand() % (8 - 2 + 1) + 2;
    }
}
```

2

Napišite funkciju nadiNajveciOpseg prototipa
void nadiNajveciOpseg(int n, struct pravokutnik* pravokutnici)

Parametar n predstavlja broj elemenata u polju pravokutnici. Vaša funkcija treba od danih pravokutnika odrediti onaj s najvećim opsegom i treba vratiti rezultat preko eksterne statičke varijable pod nazivom rjesenje (ova varijabla je već definirana tako da ju ne morate vi definirati). Navedna statička varijabla je u kodu definirana ovako:

```
struct pravokutnik rjesenje;
```

U slučaju da više pravokutnika ima maksimalan opseg, odaberite onaj koji se u polju pojavljuje ranije (onaj koji ima manji indeks).

Također deklarirate i strukturu pravokutnik koja definira pravokutnik u kartezijeovom koordinatnom sustavu preko dvije nasuprotno (nesusjedne) točke pravokutnika (npr.nastavlja se da su stranice pravokutnika paralelne s x i y osi)

My answer:

```
int abs(int broj) {
    return broj < 0 ? broj * -1 : broj;
}

struct pravokutnik {
    int x1, y1, x2, y2;
};

int vratiOpseg(int x1, int y1, int x2, int y2) {
    return 2 * abs(x2 - x1) + 2 * abs(y2 - y1);
}

void nadiNajveciOpseg(int n, struct pravokutnik* pravokutnici) {
    struct pravokutnik najveci = *pravokutnici;
    rjesenje = najveci;
    int i, najveciOpseg = vratiOpseg(najveci.x1, najveci.y1, najveci.x2, najveci.y2);
    for(i = 0; i < n; i++) {
        struct pravokutnik trenutni = *(pravokutnici + i);
        int opseg = vratiOpseg(trenutni.x1, trenutni.y1, trenutni.x2, trenutni.y2);
        if(opseg > najveciOpseg) {
            najveciOpseg = opseg;
            rjesenje = trenutni;
        }
    }
}
```

3

Napisati funkciju za izvlačenje dobite kombinacije igre Loto 6/45 prototipa:

```
void IzvlacenjeLota_6_od_45(int* kombinacija)
```

koja u jednodimenzionalno polje od 6 članova pohranjuje izvučenu kombinaciju 6 brojeva u rasponu 1-45.

Inicijalna vrijednost generatora pseudoslučajnih brojeva (pozivom funkcije srand) se inicijalizira na pozivajućoj razini (funkciji main) i ne treba je ponovno inicijalizirati.

Interval pseudoslučajnih brojeva prilagoditi traženom intervalu uz pomoć izraza koji koristi operaciju ostatka cjelobrojnog dijeljenja (prikazan na predavanjima).

Primier iedne izvučene kombinacije gdje je seed = 1. može biti na Linux sustavu (Edgar):

My answer:

```
void IzvlacenjeLota_6_od_45(int* kombinacija) {
    for(int i = 0; i < 6; i++) {
        kombinacija[i] = rand() % (45 - 1 + 1) + 1;
    }
}
```

4

1. Na početku programskog kôda kojeg ćete predati kao rješenje, izvan svih blokova, deklarirajte strukturu dionica_s kojoj su članovi definirani na sljedeći način:

- oznaka - jedinstvena oznaka dionice tipa char[] koja se sastoji od maksimalno 5 znakova (potrebno je koristiti predefinirani makro MAX_OZNAKA)
- cijena - broj tipa float
- naziv - naziv dionice char[] koji se sastoji maksimalno 20 znakova (potrebno je koristiti predefinirani makro MAX_NAZIV)

2. Osim strukture dionica_s, potrebno je definirati strukturu portfelj_s kojoj su članovi definirani na sljedeći način:

- oznaka - jedinstvena oznaka dionice tipa char[] koja se sastoji od maksimalno 5 znakova (potrebno je koristiti predefinirani makro MAX_OZNAKA)
- kolicina - cijeli broj tipa int koji predstavlja koliko posjedujemo dionica s oznakom oznaka

My answer:

```
struct dionica_s {
    char oznaka[MAX_OZNAKA];
    float cijena;
    char naziv[MAX_NAZIV];
};

struct portfelj_s
{
    char oznaka[MAX_OZNAKA];
    int kolicina;
};

float IzracunajVrijednostPortfelja(struct dionica_s *dionice, int broj_dionica, struct portfelj_s *moj_portfelj, int broj_dionica_u_mom_portfelju) {
    float vrijednost = 0;
    int i, j = 0;
    for(i = 0; i < broj_dionica; i++) {
        if(strcmp(dionice[i].oznaka, moj_portfelj[j].oznaka) == 0) {
            vrijednost += dionice[i].cijena * moj_portfelj[j].kolicina;
            j++;
        }
    }
    return vrijednost;
}
```

5

Potrebno je napisati funkciju GenerirajTelefonskiBroj koja za ulazne parametre prima:

- pokazivač broj na polje znakova tipa char

U funkciji GenerirajTelefonskiBroj potrebno je zamijeniti slovo x s pseudoslučajno generiranom znamenkom [0-9]. Ukoliko polje sadrži neki drugi znak, potrebno je taj znak zamijeniti znakom -.

Inicijalna vrijednost generatora pseudoslučajnih brojeva (pozivom funkcije srand) se inicijalizira na pozivajućoj razini (funkciji main) i ne treba je ponovno inicijalizirati.

Primjeri izvođenja funkcije koja koristi pseudoslučajno generirane brojeve:

- prva linija prikazuje vrijednost parametra broj koji ulazi u funkciju GenerirajTelefonskiBroj

My answer:

```
void GenerirajTelefonskiBroj(char *broj) {
    int i;
    char brojevi[] = {'0','1','2','3','4','5','6','7','8','9'};
    for(i = 0; broj[i] != '\0'; i++) {
        switch(broj[i]) {
            case 'x': {
                broj[i] = brojevi[rand() % (9 - 0 + 1) + 0];
                break;
            }
            default: {
                broj[i] = '-';
                break;
            }
        }
    }
}
```

6

7 bodova

Deklarirana je struktura

```
struct person_s {
    int code;
    char name[128];
    float salary;
};
```

Napisati funkciju s prototipom

My answer:

```
bool maloSlovo(char znak) {
    return znak >= 'a' && znak <= 'z' ? true : false;
}
```

```
bool velikoSlovo(char znak) {
    return znak >= 'A' && znak <= 'Z' ? true : false;
}
```

```
char pretvoriSlovo(char znak, int opcija) {
    if(opcija == 0)
        return znak + ('A' - 'a');
    return znak - ('A' - 'a');
}
```

```
bool capitalizeName(struct person_s *person) {
    if(strcmp(person->name, "") == 0) return false;
    int i;
    for(i = 0; person->name[i] != '\0'; i++) {
        char* slovo = &person->name[i];
        if(velikoSlovo(*slovo))
            *slovo = pretvoriSlovo(*slovo, 1);
    }

    for(i = 0; person->name[i] != '\0'; i++) {
        char* slovo = &person->name[i];
        if(i == 0 && maloSlovo(*slovo))
            *slovo = pretvoriSlovo(*slovo, 0);
        else if((*slovo - 1) == ' ' || *(slovo - 1) == '-') && maloSlovo(*slovo))
            *slovo = pretvoriSlovo(*slovo, 0);
    }

    return true;
}
```

7

Izračun dobitnog listića

Potrebno je napisati funkciju za obradu svih odigranih listića (pojednostavljene) igre Loto 6/45 u jednom izvlačenju.

1. Na početku programskog kôda kojeg ćete predati kao rješenje, izvan svih blokova, deklarirajte strukturu listic kojoj su članovi definirani na sljedeći način:

- serijski_broj - cijeli broj tipa long
- odigrana_kombinacija - jednodimenzionalno polje duljine 6 tipa int koje sadrži odigranu kombinaciju na listiću

Pomoću strukture listic, u programskom kôdu kojeg ne morate predati kao rješenje ovog zadatka, bit će definirana i inicijalizirana eksterna varijabla dobitni vidljiva u cijelom modulu.

My answer:

```
struct listic {
    long serijski_broj;
    int odigrana_kombinacija[6];
};
```

```
void ObradaIgre (struct listic *svi_listici, int N, int *izvuceni_brojevi) {
    int i, j;
    dobitni_serijski_broj = -1;
    for(i = 0; i < N; i++) {
        int dobitan = 1;
        for(j = 0; j < 6; j++) {
            if(svi_listici[i].odigrana_kombinacija[j] != izvuceni_brojevi[j]) {
                dobitan = 0;
                break;
            }
        }
        if(dobitan) {
            dobitni_serijski_broj = svi_listici[i].serijski_broj;
            for(j = 0; j < 6; j++)
                dobitni_odigrana_kombinacija[j] = svi_listici[i].odigrana_kombinacija[j];
            break;
        }
    }
}
```

8

Koraci za postavljanje VSC

- Kreirati direktorij D:\upro
- Konfiguracijske datoteke (dostupne u Attachments) -> c/p -> D:\upro
- Upaliti VSC, odabrati radni direktorij D:\upro
- Instalirati VSCode extension za C (detaľjan opis niže za instalaciju na fakultetskim računalima)

Upute za instalaciju Visual Studio Code ekstenzija pomoću lokalne .vsix datoteke

1. Preuzmite datoteku idiosyncrasy-compiler-extensions-64-bit-2020-08-29.vsix

My answer:

(unanswered)