

Uvod u programiranje

Ispitni rok veljača 2021/22

16 bodova

Napisati definiciju funkcije `izgladi` koja će po retcima "izgladiti" zadano dvodimenzijsko polje realnih brojeva dvostruke preciznosti.

Izgladivanje po retcima elementa 2D polja $x_{i,j}$ definiramo samo za elemente polja koji **nisu rubni (prvi i zadnji stupac)** kao:

$$smooth(x_{i,j}) = \frac{x_{i,j-1} + x_{i,j} + x_{i,j+1}}{3}.$$

Rubni elementi polja ostaju kakvi su bili.

Funkcija treba vratiti broj elemenata kojima se promijenila vrijednost.

Primjer 1:

Zadano 2x5 polje:

```
1.00 1.00 0.00 0.00 1.00
1.00 1.00 1.00 0.00 1.00
```

treba promijeniti u:

```
1.00 0.67 0.33 0.33 1.00
1.00 1.00 0.67 0.67 1.00
```

pri čemu treba vratiti `5`.

Također, ovdje primijetite kako je izračunat treći element prvog reda, kao:

$0.33 = \frac{1 + 0 + 0}{3}$ (uvijek se koristi originalna vrijednost matrice za izračun)

Analogno za četvrti element, itd.

Primjer 2:

Zadano 1x3 polje:

```
0.00 0.00 3.00
```

treba promijeniti u:

```
0.00 1.00 3.00
```

pri čemu treba vratiti `1`.

Napomena: Kao rješenje dostaviti samo definiciju funkcije `izgladi`, odnosno dio programskog koda koji se u gore prikazanom modulu nalazi između `/* POCETAK DEFINICIJE */` i `/* KRAJ DEFINICIJE */`. `m` je broj redaka, a `n` broj stupaca.

```
#include <stdio.h>

/* POCETAK DEFINICIJE */
// m - broj redaka matrice
ovdje_navesti_tip_funkcije izgladi(ovdje_navesti_prvi_parametar, int m, int n) {
    ovdje_napisati_tijelo_funkcije
}
/* KRAJ DEFINICIJE */

int main(void) {
    ...
}
```

Student's answer:

```
1 int izgladi(double *p, int m, int n) {
2     int promjena = 0;
3
4     for(int i = 0; i < m; i++) {
5         double polje[n];
6
7         for(int j = 0; j < n; j++) {
8             polje[j] = *(p + n*i + j);
9         }
10
11        for(int j = 1; j < (n-1); j++) {
12            *(p + n*i + j) = (polje[j-1] + polje[j] + polje[j+1]) / 3;
13
14            if(*(p + n*i + j) != polje[j]) {
15                promjena++;
16            }
17        }
18    }
19
20    return promjena;
21 }
```

Hint: Correct. Well done!

16 bodova

Napisati **funkciju** unosKonverzija tipa int koja kao parametre prima redom:

- pokazivač na prvi član niza znakova (string) : niz
- pokazivač na objekt tipa int : dulj

Funkcija unosKonverzija sa standardnog ulaza učitava niz znakova dok se ne unese znak uskličnik (!) koji ne ulazi u učitani niz ili dok duljina učitnog niza nije veća od **10 znakova**. Dozvoljen je unos znakova razmaka. Tijekom učitavanja funkcija konvertira sva velika slova u mala i sve znamenke u znak upitnik (?). Konvertirani i ispravno terminirani niz vraća se kao parametar niz, a duljina učitnog niza, bez znaka za kraj niza, kao parametar dulj. Broj konvertiranih znamenki funkcija treba vratiti preko imena u pozivajući program.

Napomena: potrebno je predati samo izvorni kod funkcije unosKonverzija. Funkcija ništa ne ispisuje.

Primjeri ispravnih rezultata (unos od korisnika: niz; ispis: konvertiran niz, duljina dulj, rezultat poziva funkcije unosKonverzija). Okomite crte u ispisu samo vizualiziraju početak i kraj konvertiranog niza:

Unos:

12·34·5·!

Ispis:

|??·??·?·| 8 5

Unos:

1A··B23·c4!

Ispis:

|?a··b??·c?| 10 4

Unos:

Upro1234567890!

Ispis:

|upro??????| 10 6

Student's answer:

```
1 int unosKonverzija(char *niz, int *dulj) {
2     char zn;
3     int n = 0, konv = 0;
4
5     while(n < 10 && zn != '!') {
6         scanf("%c", &zn);
7
8         if(zn != '!') {
9             if(zn >= 'A' && zn <= 'Z') {
10                 zn = zn + ('a' - 'A');
11             }
12             else if(zn >= '0' && zn <= '9') {
13                 zn = '?';
14                 konv++;
15             }
16             *(niz + n) = zn;
17         } else {
18             break;
19         }
20
21         n++;
22     }
23
24     *(niz + n) = '\\0';
25     *dulj = n;
26
27     return konv;
28 }
```

Hint: Correct. Well done!

16 bodova

Napisati funkciju s prototipom

```
void put(int seed, int koraka, struct tocka_s *tocke)
```

koja puni polje na čiji prvi član pokazuje pokazivač `tocke`. Polje se puni na sljedeći način:

- za prvu točku postavi se ishodište
- koordinate sljedećih točaka izračunavaju se tako da se na koordinatu prethodne točke doda slučajno izračunati pomak po X koordinati (nova slučajna vrijednost u rasponu od 0 do 10) i slučajno izračunati pomak po Y koordinati (druga nova slučajna vrijednost u rasponu od 0 do 10).

Broj `seed` je inicijalna vrijednost za generator pseudoslučajnih brojeva, a broj `koraka` predstavlja broj točaka dodanih nakon točke ishodišta (minimalno jednak 1).

Deklaracija strukture je

```
struct tocka_s {  
    int x;  
    int y;  
};
```

Važno: za transformaciju pseudoslučajnog broja u broj iz željenog intervala koristiti operator `%`.

Primjer 1:

Za zadano sjeme `500` i broj koraka `2` funkcija će u polje dodati točke `(0,0)`, `(10,6)` i `(14,15)` (ako testirate na operacijskom sustavu Windows), odnosno `(0,0)`, `(10,3)` i `(18,6)` (ako testirate na operacijskom sustavu Linux ili u sustavu Edgar).

Primjer 2:

Za zadano sjeme `1234` i broj koraka `1` funkcija će u polje dodati točke `(0,0)` i `(9,4)` (ako testirate na operacijskom sustavu Windows), odnosno `(0,0)` i `(3,7)` (ako testirate na operacijskom sustavu Linux ili u sustavu Edgar).

Predati samo definiciju funkcije, bez naredbi `#include`, definicije strukture i glavnog programa.

Student's answer:

```
1 void put(int seed, int koraka, struct tocka_s *tocke) {
2     int i;
3     srand((unsigned int)seed);
4
5     (tocke + 0)->x = 0;
6     (tocke + 0)->y = 0;
7
8     for(i = 1; i <= koraka; i++) {
9         (tocke + i)->x = (tocke + i - 1)->x + rand() % 11;
10        (tocke + i)->y = (tocke + i - 1)->y + rand() % 11;
11    }
12
13    return;
14 }
```

Hint: Correct. Well done!

16 bodova

Načiniti funkciju s prototipom `_Bool setQuadrants(char *fileName)` koja u binarnoj datoteci zadanog imena svakoj točki ažurira kvadrant u kojem se nalazi. Kvadranti su numerirani od 1 do 4, od "sjeveroistočnog", u pravcu suprotnom kazaljci na satu. Neće se provjeravati rubni uvjeti (u testnim primjerima neće biti točaka na koordinatnim osima). Funkcija treba vratiti 1 ako je obavljena uspješno, a 0 ako nije uspjelo otvaranje toka.

Zapisi u `file` su oblika

```
struct record_s {  
    int x;  
    int y;  
    int quadrant;  
};
```

Važno: Nakon svakog upisivanja, pozvati `fflush`.

Napomena: predati samo izvorni kod funkcije, bez deklaracije strukture i naredbi `#include`. Glavni program, deklaracija strukture i naredbe `#include` kojima se testira funkcionalnost funkcije već su pripremljeni u Edgaru. U funkciji obvezno zatvoriti tok, jer inače načinjene promjene neće biti spremljene ni vidljive programu koji testira funkciju. **Nije dozvoljeno korištenje polja.**

Primjer ulazne datoteke nalazi se u privitku.

Student's answer:

```
1 _Bool setQuadrants(char *fileName) {
2     FILE *bin = fopen(fileName, "r+b");
3     if(bin == NULL) {
4         return 0;
5     }
6
7     struct record_s record;
8
9     while(fread(&record, sizeof(record), 1, bin) == 1) {
10         if(record.x > 0 && record.y > 0) {
11             record.quadrant = 1;
12         }
13         else if(record.x < 0 && record.y > 0) {
14             record.quadrant = 2;
15         }
16         else if(record.x < 0 && record.y < 0) {
17             record.quadrant = 3;
18         } else {
19             record.quadrant = 4;
20         }
21
22         fseek(bin, -1L*sizeof(record), SEEK_CUR);
23         fwrite(&record, sizeof(record), 1, bin);
24     }
25
26     fclose(bin);
27
28     return 1;
29 }
```

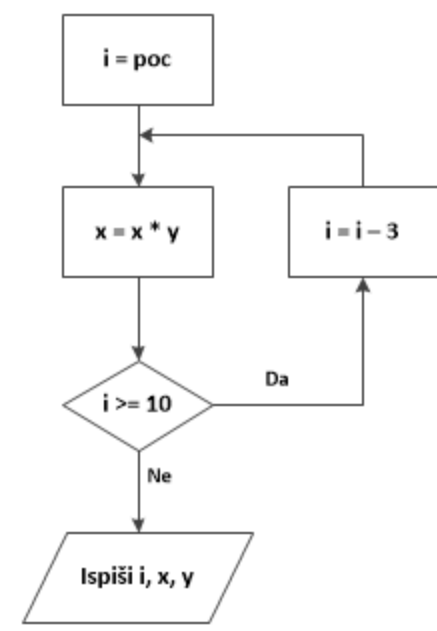
Hint: Correct. Well done!

4 boda

Napisati odsječak programa koji, korištenjem **petlje s ispitivanjem uvjeta na kraju**, obavlja potpuno istu funkciju kao odsječak programa prikazan dijagramom toka. Nije dozvoljeno korištenje pomoćnih varijabli.

```
#include <stdio.h>

int main(void) {
    int i, poc;
    float x, y;
    scanf("%d %f %f", &poc, &x, &y);
```



```
    return 0;
}
```

Ako konačne vrijednosti varijabli `i`, `x` i `y` redom iznose 9, 10567.2 i 1.5 tada ispis treba biti sljedećeg oblika:

```
i = 9
x = 10567.20
y = 1.50
```

Napomena: početak i završetak programa prikazan u tekstu zadatka nije potrebno pisati.

Student's answer:

```
1 {  
2     i = poc;  
3     do {  
4         x = x * y;  
5         if(i < 10) {  
6             break;  
7         }  
8         i = i - 3;  
9     } while(1);  
10  
11     printf("i = %d\nx = %.2f\ny = %.2f", i, x, y);  
12 }
```

Hint: Correct. Well done!

4 boda

Na mjestu u programu označenom s `/* Ovdje napisati naredbu s jednim pozivom funkcije scanf */` napisati **točno jednu** naredbu (točno jedan poziv funkcije `scanf`).

```
#include <stdio.h>
#define MAXNIZ 40
int main(void) {
    double x;
    int m;
    char znak;
    char niz[MAXNIZ + 1];

    /* Ovdje napisati naredbu s jednim pozivom funkcije scanf */

    printf("%f|%d|%c|%s|\n", x, m, znak, niz);

    return 0;
}
```

Pozivom funkcije se za sljedeće podatke upisane preko tipkovnice

```
2.25-21↵
...E...G↵
.....Tom Sawyer!!!↵
```

u varijable `x`, `m`, `znak` i `niz` trebaju učitati vrijednosti tako da program na zaslon ispiše

```
2.250000|-21|E|Tom Sawyer|↵
```

Napomena: Kao rješenje dostaviti točno **jednu** naredbu s pozivom funkcije `scanf` (naredbu obavezno terminirati znakom točka-zarez). Definiranje dodatnih varijabli nije dopušteno. Znak `·` u prikazu podataka upisanih preko tipkovnice predstavlja prazninu.

Student's answer:

1 scanf("%lf%d %c %*c %[Tom Sawyer\n]", &x, &m, &znak, niz);

Correct answer:

1 This test does not have "show solutions" option enabled.

4 boda

Ako se vrijednost neke varijable upisuje u tekstnu datoteku, broj upisanih bajtova, u odnosu na broj bajtova koji varijabla zauzima u središnjoj memoriji

a je uvijek manji

b je uvijek veći

c je uvijek jednak

d može biti manji, jednak ili veći

4 boda

Napisati **jednu** naredbu koja će, ugrađena u prikazani odsječak programa na mjestu označenom s `XXXXXXXXXX`, u varijablu `predmet` tipa `predmet_t` postaviti za ECTS bodove vrijednost **7** tako da nakon izvršavanja kompletnog programa kao rezultat bude ispisano:

```
500·Uvod·u·programiranje·7
```

```
#include <stdio.h>
int main (void) {
    typedef struct {
        int sifPred;
        char nazPred[20 + 1];
        int ECTSBod;
    } predmet_t;
    predmet_t predmet = {500, "Uvod u programiranje", 0}, *p_predmet = &predmet;
    {
        int predmet = 9;
        XXXXXXXXXXXX
    }
    printf("%d %s %d", predmet.sifPred, predmet.nazPred, predmet.ECTSBod);
    return 0;
}
```

Napomena: Kao rješenje dostaviti samo **jednu** naredbu (naredbu obavezno terminirati znakom točka-zarez).

Student's answer:

```
1 p_predmet->ECTSBod = 7;
```

Correct answer:

```
1 This test does not have "show solutions" option enabled.
```

Hint: Correct. Well done!