

Uvod u programiranje

Završni ispit 2021/22

prvi termin

9 bodova

Napisati funkciju *odrediMjesecIDanUskrsa* tipa *void* koja kao parametre prima redom:

- `godina` tipa `int`, za kojega možete pretpostaviti da će biti četveroznamenasti pozitivni broj iz intervala [1582, 2099] (nije potrebno provjeravati)
- `mjesec` tipa pokazivač na objekt tipa `int`
- `dan` tipa pokazivač na objekt tipa `int`

Funkcija *odrediMjesecIDanUskrsa* u pozivajući program preko parametara `mjesec` i `dan`, za zadanu godinu (`godina`) treba vratiti mjesec i dan Uskrsa izračunatog prema Gaussovom algoritmu (opisan u nastavku).

Za izračun mjeseca i dana Uskrsa koristite X i Y, čija vrijednost ovisi o godini.

Vrijednosti X i Y su dane u priloženoj tablici:

godina	X	Y
1582. -1699.	22	2
1700. – 1799.	23	3
1800. -1899.	23	4
1900. -2099.	24	5

Pomoću X i Y izračunaju se sljedeće vrijednosti prema formulama:

A = ostatak dijeljenja godine s 19

B = ostatak dijeljenja godine s 4

C = ostatak dijeljenja godine sa 7

D = ostatak dijeljenja (A \* 19 + X) s 30

E = ostatak dijeljenja (2 \* B + 4 \* C + 6 \* D + Y) sa 7

Prema Gaussovom algoritmu, datum Uskrsa je (22 + D+ E). **ožujka** ako (22 +D + E) ne prelazi 31, inače je (D + E -9). **travnja**.

Za kontrolu ispravnosti rada funkcije u nastavku su datumi Uskrsa za nekoliko godina:

1603	1950	2050
30.3.1603	9.4.1950	10.4.2050

Kao rješenje predati samo definiciju funkcije, bez naredbi `#include` i glavnog programa.

Correct answer:

```
1 void odrediMjesecIDanUskrsa(int godina, int *mjesec, int *dan) {
2     int X, Y, A, B, C, D, E;
3
4     if (godina >= 1582 && godina <= 1699){
5         X = 22;
6         Y = 2;
7     }
8     if (godina >= 1700 && godina <= 1799) {
9         X = 23;
10        Y = 3;
11    }
12    if (godina >= 1800 && godina <= 1899) {
13        X = 23;
14        Y = 4;
15    }
16    if (godina >= 1900 && godina <= 2099) {
17        X = 24;
18        Y = 5;
19    }
20    A = godina % 19;
21    B = godina % 4;
22    C = godina % 7;
23    D = (A * 19 + X) % 30;
24    E = (2 * B + 4 * C + 6 * D + Y) % 7;
25    if ((22 + D + E) <= 31) {
26        *mjesec = 3;
27        *dan = 22 + D + E;
28    } else {
29        *mjesec = 4;
30        *dan = D + E - 9;
31    }
32    return;
33 }
```

## 9 bodova

Napisati definiciju funkcije `pokNaNajblizuTocku` koja u zadanom jednodimenzijskom polju od `n` članova polja tipa `tocka_t` pronalazi točku koja je najbliža ishodištu pravokutnog koordinatnog sustava, a zatim preko imena funkcije u pozivajući program vraća pokazivač na dotičnu točku (tj. pokazivač na pronađeni član polja).

Ako u polju postoji više točaka koje imaju jednaku najmanju udaljenost do ishodišta, funkcija u pozivajući program treba vratiti pokazivač na *posljednju* od takvih točaka u polju.

Udaljenost između točaka  $(x_1, y_1)$  i  $(x_2, y_2)$  u pravokutnom koordinatnom sustavu izračunava se pomoću izraza  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ .

**Primjer 1:** U zadanom polju točaka `(0.7, 1.2), (0.7, 1.1), (-1.0, 1.0), (-1.1, -1.0)`, druga točka je najbliža ishodištu, što znači da funkcija u pozivajući program treba vratiti pokazivač na drugi član polja.

**Primjer 2:** U zadanom polju točaka `(1.5, 2.5), (1.7, 2.6), (-2.5, -1.5), (1.9, -3.1), (2.1, -4.1)`, ishodištu su najbliže (jednako blizu) prva i treća točka, što znači da funkcija u pozivajući program treba vratiti pokazivač na treći član polja.

**Primjer 3:** U zadanom polju točaka `(2.e+121, 2.e+121), (1.e+121, 1.e+121), (4.e+121, 4.e+121)`, druga točka je najbliža ishodištu, što znači da funkcija u pozivajući program treba vratiti pokazivač na drugi član polja.

```
#include <stdio.h>
#include <math.h>

typedef struct {
    double x;
    double y;
} tocka_t;

/* POCETAK DEFINICIJE */
ovdje_navigate_tip_funkcije pokNaNajblizuTocku(ovdje_navigate_prvi_parametar, int n) {
    ovdje_napisati_tijelo_funkcije
}
/* KRAJ DEFINICIJE */

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

**Napomena:** Kao rješenje dostaviti samo definiciju funkcije `pokNaNajblizuTocku`, odnosno dio programskog koda koji se u gore prikazanom modulu nalazi između `/* POCETAK DEFINICIJE */` i `/* KRAJ DEFINICIJE */`.

Correct answer:

```
1  tocka_t *pokNaNajblizuTocku(tocka_t *polje, int n) {
2      int indTrazeneTocke;
3      double najmUdalj;
4      for (int i = 0; i < n; ++i) {
5          double udalj = sqrt(pow((polje + i)->x, 2.) + pow((polje + i)->y, 2.));
6          if (i == 0 || udalj <= najmUdalj) {
7              indTrazeneTocke = i;
8              najmUdalj = udalj;
9          }
10     }
11     return polje + indTrazeneTocke;
12 }
```

## 9 bodova

Napisati definiciju funkcije `generirajLozinku` prototipa

```
void generirajLozinku (unsigned int sjeme, const char *znakoviZaLozinku, int duljLoz, char *lozinka);
```

koja kao parametre prima redom:

- inicijalnu vrijednost za generator pseudoslučajnih brojeva
- pokazivač na prvi član niza znakova (*string*) koji treba koristiti pri generiranju lozinke
- duljinu lozinke koju je potrebno generirati
- pokazivač na prvi član niza znakova (*string*) kojeg treba popuniti zadanim brojem nasumično odabranih znakova

Funkcija treba generirati lozinku `lozinka` duljine `duljLoz` tako da pomoću generatora pseudoslučajanih brojeva odabire znakove sadržane u nizu znakova `znakoviZaLozinku`.

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

### Primjer 1:

Za zadano sjeme `500`, niz `123456789qwertyuiopasdfghjklxyxcvbnmQWERTZUIOPASDFGHJKLYXCVBNM#$%&()'` i duljinu `10` funkcija generira lozinku `dH5LHZqr&I` (ako testirate na operacijskom sustavu Windows), odnosno `w4azVBdI`)p (ako testirate na operacijskom sustavu Linux ili u sustavu Edgar).

### Primjer 2:

Za zadano sjeme `7351`, niz `ABC-123` i duljinu `15` funkcija generira lozinku `2--AA2-BC2-BAB-` (ako testirate na operacijskom sustavu Windows), odnosno `C1BA3B-31C2BC2B` (ako testirate na operacijskom sustavu Linux ili u sustavu Edgar)

---

Kao rješenje predati samo definiciju funkcije, bez naredbi `#include` i glavnog programa.

Correct answer:

```
1 void generirajLozinku(int sjeme, char *znakoviZaLozinku, int duljLoz, char *lozinka){
2     int i, dulj;
3
4     srand((unsigned)sjeme);
5     dulj = strlen(znakoviZaLozinku);
6     for (i = 0; i < duljLoz; i++)
7         *(lozinka + i) = znakoviZaLozinku[(rand() % dulj)];
8     *(lozinka + i) = 0;
9     return;
10 }
11
```

## 9 bodova

U nekoj binarnoj datoteci nalaze se podaci o plaćama osoba. Zapisi su u obliku:

```
struct record {  
    int code;  
    int salary;  
    char name[40 + 1];  
};
```

Napisati funkciju s prototipom

```
void increaseLowest(char *fileName, int percentage)
```

koja u datoteci zadanog imena poveća najnižu plaću za zadani postotak (nije potrebno zaokruživati rezultat).  
Ako više osoba ima jednaku najnižu plaću, povećati plaću onoj s najmanjim rednim brojem.

Redni broj zapisa odgovara šifri osobe. Zapis sa šifrom 1 nalazi se na samom početku datoteke.

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

Primjer binarne datoteke nalazi se u privitku.



Correct answer:

```
1 void increaseLowest(char *name, int percentage) {
2     struct record r;
3     FILE *f;
4     f = fopen(name, "r+b");
5     int pos = 0, posMin, minSalary;
6     while (fread(&r, sizeof(r), 1, f) == 1) {
7         if (pos == 0 || r.salary < minSalary) {
8             posMin = pos;
9             minSalary = r.salary;
10        }
11        pos++;
12    }
13    fseek(f, posMin * sizeof(r), SEEK_SET);
14    fread(&r, sizeof(r), 1, f);
15    r.salary *= 1 + percentage/100.;
16    fseek(f, posMin * sizeof(r), SEEK_SET);
17    fwrite(&r, sizeof(r), 1, f);
18    fclose(f);
19 }
```

### 3 boda

Podaci su pohranjeni u strukturi

```
struct podaci_s {  
    char prezime[40 + 1];  
    char ime[40 + 1];  
    float primanja;  
};
```

Funkcija prototipa:

```
char *trazenoPrezime(struct podaci_s *pok, int n);
```

kao parametre prima pokazivač na prvi član polja struktura `podaci_s` i cjelobrojnu varijablu `n`, a u pozivajući program preko imena funkcije vraća `prezime` iz člana polja s indeksom `n`. Čime u donjem programskom odsječku treba zamijeniti IZRAZ, kako bi funkcija ispravno vratila traženi podatak?

```
char *trazenoPrezime(struct podaci_s *pok, int n) {  
    return IZRAZ;  
}
```

Student's answer:

```
1 (pok + n)->prezime
```

Hint: Correct. Well done!

Correct answer:

`((n+pok)).prezime` ili  
`((n+pok)).prezime`) ili  
`((pok+n)).prezime` ili  
`((pok+n)).prezime`) ili  
`(n+pok)->prezime` ili  
`((n+pok)->prezime)` ili  
`(pok+n)->prezime` ili  
`((pok+n)->prezime)` ili  
`pok[n].prezime` ili `pok[n].prezime`)

## 2 boda

Koliko će byteova u memoriji zauzeti sljedeća definicija polja?

```
char p[][7] = {  
    { 88, 97, 41, 44, 18, 65, 37 },  
    { 35, 1, 2, 10, 75, 40 },  
    { 49, 18, 97, 58, 13 }  
};
```

Ako smatrate da je definicija neispravna ili da se ne može jednoznačno odgovoriti na ovo pitanje, kao odgovor upišite .

Student's answer:

1 21

Hint: Correct. Well done!

Correct answer:

42.

## 2 boda

### Funkcija

```
char *broji (char *niz, int *brVel, int *brMal);
```

vraća pokazivač na prvi samoglasnik u zadanom nizu te također izračunava i preko parametara vraća broj velikih slova (`brVel`) i broj malih slova (`brMal`) u zadanom nizu.

Čime u donjem programskom odsječku treba zamijeniti XXXXXXXXXX, kako bi se naredbom `printf` ispisao prvi samoglasnik, broj velikih slova (`velika`) i broj malih slova (`mala`) u zadanom nizu?

```
int main(void) {  
    int velika, mala;  
    char *p = NULL;  
    char niz[MAX_NIZ + 1];  
    fgets(niz, MAX_NIZ + 1, stdin);  
    XXXXXXXXXX  
    if (p == NULL) {  
        printf("U nizu nema samoglasnika.");  
    } else {  
        printf("%c\n", *p);  
    }  
    printf("Velikih: %d\n", velika);  
    printf("Malih: %d\n", mala);  
    return 0;  
}
```

### Student's answer:

```
1 p = broji(niz, &velika, &mala);
```

Hint: Correct. Well done!

## 2 boda

Datoteka čijim se zapisima treba pristupiti direktno, znajući samo redni broj zapisa, mora biti:

**a** isključivo binarna, pri čemu zapisi moraju biti jednake doljine

**b** isključivo binarna, pri čemu zapisi mogu biti različitih doljina

**c** isključivo tekstna, pri čemu zapisi moraju biti jednake doljine

**d** isključivo tekstna, pri čemu zapisi mogu biti različitih doljina

**e** tekstna ili binarna, pri čemu zapisi moraju biti jednake doljine