

JMBAG		Prezime, ime	
-------	--	--------------	--

Uvod u programiranje

Međuispit - 16. studenog 2020.

Napomene:

- Programska rješenja moraju biti usklađena sa stilom pisanja programa koji je propisan na predmetu
- Rješenja zadataka 1 - 4 napisati na vlastitim papirima i predati ih u košuljici
- Rješenja zadataka 5 - 6 napisati na za to predviđeno mjesto uz tekst zadatka te list sa zadacima predati u košuljici

1. (10 bodova)

Napisati funkciju `flipCase` koja kao parametar prima jedan znak `c (char)`. Funkcija u pozivajući program treba vratiti jedan znak (`char`): ako je `c` malo slovo, funkcija vraća odgovarajuće veliko slovo, ako je `c` veliko slovo, funkcija vraća odgovarajuće malo slovo, a ako znak `c` nije slovo, funkcija vraća taj isti znak `c`.

U nastavku napisati funkciju `main` ("glavni program") u kojoj s tipkovnice treba učitati niz znakova koji zajedno s eventualno učitanim oznakom novog reda sigurno neće biti dulji od 30 znakova. Zatim promijeniti sadržaj učitano niza: uz korištenje funkcije `flipCase`, sva mala slova u nizu pretvoriti u velika, a sva velika slova u mala, te nakon toga dobiveni novi sadržaj niza ispisati pomoću konverzijske specifikacije `%s`.

Primjeri izvršavanja programa

```
Upisite niz > Medjuispit 2020!
Rezultat >> mEDJUISPIT 2020!
```

```
Upisite niz > 24*5, 345, ###?!
Rezultat >> 24*5, 345, ###?!

```

2. (10 bodova)

Učitati pozitivne realne brojeve a , b i prirodni broj n . Nije potrebno provjeravati ispravnost učitanih podataka. Brojevi a i b predstavljaju početne članove nizova A i B (a_1 odnosno b_1). Za sve $i > 1$, član niza a_i izračunava se kao aritmetička sredina prethodnog člana niza A i prethodnog člana niza B , tj. $a_i = \frac{a_{i-1} + b_{i-1}}{2}$. Član niza b_i izračunava se kao geometrijska sredina prethodnog člana niza A i prethodnog člana niza B , tj. $b_i = \sqrt{a_{i-1} * b_{i-1}}$. Članove nizova A i B , od a_1 do a_n , te b_1 do b_n ispisati u skladu s oblikom ispisa prikazanim u primjeru. U programu nije dopušteno koristiti agregatne tipove podataka.

Primjer izvršavanja programa

```
Upisite a, b, n > 1 2000 5
A(1)= 1.000 B(1)=2000.000
A(2)=1000.500 B(2)= 44.721
A(3)= 522.611 B(3)= 211.527
A(4)= 367.069 B(4)= 332.485
A(5)= 349.777 B(5)= 349.349
```

3. (10 bodova)

Brojevi bankovnih računa (`int`) na koje se uplaćuju iznosi u kunama (`int`) su iz intervala $[100000, 101000]$. Na početku iznosi na svim računima trebaju biti 0 kuna. Pretpostavlja se da će uplaćeni iznosi biti cijeli brojevi koji predstavljaju kune (tj. nisu dopuštene uplate u lipama).

Učitavati parove brojeva: broj računa i iznos uplate u kunama. Korisnik će sigurno upisati barem jedan par brojeva. Učitavanje prekinuti onda kada broj uplata na bilo kojem od računa dosegne 3 uplate. Nakon toga ispisati najveću sumu uplata po bilo kojem računu. Ispis mora biti usklađen s oblikom ispisa prikazanim u primjeru.

U primjeru: učitavanje je prekinuto jer su na račun 100015 stigle 3 uplate. Najveća suma uplata po bilo kojem računu je 250 kuna (nije važno što se ista najveća suma nalazi i na računu 100002 i na računu 100550).

Primjer izvršavanja programa

```
Upisajte uplate >
100015 10
100006 170
100001 150
100002 230
100550 250
100015 77
100002 20
100015 30
Najveća suma: 250 kn
```

Sljedeći programski odsječak realizirati bez naredbi continue, break i goto, te bez definiranja i korištenja dodatnih varijabli.

```

z = x;
labela_A:
z = (z + w) / 1.2;
i = i - t;
if (i < m) {
    k = k + 100;
} else {
    k = k + 1000;
    goto labela_B;
}
if (z >= 200.) goto labela_A;
labela_B:
z = z * 1000.;

```

5. (3 bodq)

U skladu sa standardom IEEE 754 za prikaz brojeva standardne preciznosti, u registru je pohranjen negativni realni broj $-10110.11_2 \cdot 2^{-133}$. Sadržaj tog registra prikažite u **oktalnom** brojevnom sustavu (potrebno je napisati samo konačni rezultat u obliku oktalnog broja, postupak ne treba dostaviti u rješenju).

Prostor za odgovor

6. (4 boda)

Prikazati što će se na zaslonu ispisati tijekom izvršavanja odsječka programa, vodeći računa o točnoj poziciji ispisanih znakova: svaki "kvadratić" na papiru predstavlja mjesto na zaslonu na koje se ispisuje po jedan znak; svaki "redak kvadratića" predstavlja jedan redak na zaslonu.

[illegible]

```
double x = 9.;
int y;
y = (int) x / 2+1.0;
printf("%d\n", y);

unsigned int i = 012;
printf("%u\n", i & i << 2);

char a[10] = {'0', 10, '1', '2', 0, 1, 2};
printf("%s\n", a);

int m = 15, n = 5, k;
k = m++ - ++n;
printf("%d %d %d", k, m, n);
```

Rješenja

```
1. #include <stdio.h>
#define MAX_NIZ 30
char flipCase(char c) {
    if (c >= 'A' && c <= 'Z') {
        c = c + ('a' - 'A');
    } else if (c >= 'a' && c <= 'z') {
        c = c - ('a' - 'A');
    }
    return c;
}
```

```
int main(void) {
    char niz[MAX_NIZ + 1];
    printf("Upisite niz > ");
    fgets(niz, MAX_NIZ + 1, stdin);
    int i = 0;
    while (niz[i] != '\0') {
        niz[i] = flipCase(niz[i]); // dobro je ovako, a dobro je i ako se
        i = i + 1;                // prije izmjene pita radi li se o slovu
    }
    printf("Rezultat > %s", niz);
    return 0;
}
```

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

int main(void) {
    float a_i, b_i, a_i_minus1, b_i_minus1;
    int i, n;
    printf("Upisite a, b, n > ");
    scanf("%f %f %d", &a_i, &b_i, &n);
    for (i = 1; i <= n; ++i) {
        if (i > 1) {
            a_i = (a_i_minus1 + b_i_minus1) / 2.f;
            b_i = sqrt(a_i_minus1 * b_i_minus1);
        }
        printf("A(%d)=%8.3f B(%d)=%8.3f\n", i, a_i, i, b_i);
        a_i_minus1 = a_i;
        b_i_minus1 = b_i;
    }
    return 0;
}
```

```
ALT:
char rez;
if (c >= 'A' && c <= 'Z') {
    rez = c + ('a' - 'A');
} else if (c >= 'a' && c <= 'z') {
    rez = c - ('a' - 'A');
} else {
    rez = c;
}
return rez;
```

