

Ime i prezime: _____ Broj indeksa: _____ / _____

1.(12b) Korisnik zadaje niz realnih brojeva a dužine n . Napisati program koji određuje indeks maksimalnog elementa niza a koji je veći od srednje vrednosti tog niza.

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

```
int main()
{
    int n, i_maks;
    double a[50], sr = 0;
    printf("Uneti duzinu niza: ");
    scanf_s("%d", &n);
    printf("Uneti elemente niza:\n");
    for (int i = 0; i < n; i++)
    {
        printf("a[%d]= ", i);
        scanf_s("%lf", &a[i]); //Ucitavanje elemenata niza a;
        sr = sr + a[i]; //Sabiranje svih elemenata niza a;
    }
    sr /= n; //Racunanje srednje vrednosti niza;
    double maks = a[0]-1;
    for (int i = 0; i < n; i++)
    {
        if (a[i] > sr && a[i] > maks)
        {
            maks = a[i];
            i_maks = i;
        }
    }
    printf("Indeks maksimalnog elementa veceg od sr. vrednosti niza: %d.\n",
i_maks);
}
```

2.(12b) Napisati program koji za tekst koji se unosi sa tastature znak po znak određuje minimalnu dužinu reči koje počinju velikim slovom, kao i broj tih reči. Smatrati da su reči razdvojene blanko razmacima, prelaskom u novi red i horizontalnom tabulacijom.

```
#include <stdio.h>
#include <ctype.h> //Zbog isupper();

int main()
{
    char c, p = ' '; //c - trenutni znak, p - prethodni znak;
    int broj = 0, n, d, min = 1E10;
    while ((c = getchar()) != EOF)
    {
        //Na pocetku smo nove reci i ona pocinje velikim slovom:
        if ((p == ' ' || p == '\n' || p == '\t') && (isupper(c)))
        {
            broj++;
            n = 1;
        }
        //Na pocetku smo nove reci i ona ne pocinje velikim slovom:
        else if ((p == ' ' || p == '\n' || p == '\t') && (!isupper(c))) n = 0;
        //Ako je nova rec sa velikim slovom racunamo njenu duzinu, n:
        if (n) n++;
        //Odredjivanje min. duzine reci sa velikim slovom:
        if ((c == ' ' || c == '\n' || c == '\t') && (n < min)&&(n>0)) min = n;
        p = c;
    }
    printf("Broj takvih reci je: %d, a minimalna duzina iznosi: %d.\n", broj,
min-2);
}
```

3.(13b) Korisnik zadaje niz realnih brojeva ***a*** dužine ***n***. Napisati funkciju koja modifikuje niz ***a*** tako da mu elementi postanu indeksi onih elemenata niza ***a*** čija je vrednost jednaka minimalnoj vrednosti niza.

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

void izmena(float a[], int &n)
{
    float min = INT_MAX;
    //Odredjivanje minimuma, min:
    for (int i = 0; i < n; i++)
    {
        if (a[i] < min) min = a[i];
    }
    //Odredjivanje indeksa minimalnih elemenata:
    int j = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] == min) a[j++] = i;
    }
    n = j;
}

//Testiranje (nije trazeno u zadatku):

int main()
{
    int n;
    float a[50];
    printf("Uneti duzinu niza: ");
    scanf_s("%d", &n);
    printf("Uneti niz:\n");
    for (int i = 0; i < n; i++)
    {
        printf("a[%d] = ", i);
        scanf_s("%f", &a[i]);
    }
    izmena(a, n);
    printf("Nove vrednosti niza:\n");
    for (int i = 0; i < n; i++)
        printf("a[%d] = %.2f\n", i, a[i]);
}
```

4.(13b) Potrebno je napisati funkciju **sinusi** koja računa sinuse dva ugla pravouglog trougla. Ulaz u funkciju je struktura **trougao** koju čine koordinate temena pravouglog trougla, koju treba zasebno definisati. Smatrati da je prvo teme u strukturi **trougao** teme pravog ugla.

```
#include <stdio.h>
#include <math.h> //Zbog pow();
```

```
struct trougao
```

```
{
    float xa, ya, xb, yb, xc, yc;
};
```

```
void sinusi(trougao t, float &sina, float &sinb)
```

```
{
    float a, b, c;
    a = sqrt(pow(t.xb - t.xc, 2) + pow(t.yb - t.yc, 2));
    b = sqrt(pow(t.xa - t.xc, 2) + pow(t.ya - t.yc, 2));
    c = sqrt(pow(t.xa - t.xb, 2) + pow(t.ya - t.yb, 2));
    sina = b / a;
    sinb = c / a;
}
```

```
//Trestiranje (nije trazeno u zadatku):
```

```
int main()
{
    trougao t;
    float sinalfa, sinbeta;
    printf("Uneti koordinate prav. trougla: ");
    scanf_s("%f%f%f%f%f", &t.xa, &t.ya, &t.xb, &t.yb, &t.xc, &t.yc);
    sinusi(t, sinalfa, sinbeta);
    printf("Sinusi uglova su: %.2f i %.2f.\n", sinalfa, sinbeta);
}
```

