

Ime i prezime: \_\_\_\_\_ Broj indeksa: \_\_\_\_\_ / \_\_\_\_\_

**1.(10b)** Korisnik zadaje niz prirodnih brojeva  $a$  sa brojem elemenata  $n$ . Pri unosu svakog elementa niza treba da se pojavi oznaka elementa (npr.  $a[2] =$ ). Napisati program koji izračunava proizvod elemenata niza koji prethode prvom elementu sa vrednosti 100.

```
#include <stdio.h>

int main()
{
    int n, a[50], p = 1, k = 1;
    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("%d", &a[i]);
        if ((a[i] != 100) && k) p *= a[i];
        else k = 0;
    }
    printf("Proizvod: %d.\n", p);
}
```

**2.(12b)** Napisati program koji za uneti string  $s$  izračunava prosečnu dužinu onih reči čija je dužina veća od 3, a manja od 7 znakova (reči su razdvojene razmacima).

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

int main()
{
    char s[50];
    int d = 0, n = 0;
    float sr = 0;
    printf("Uneti string:\n");
    gets_s(s);
    strcat_s(s, " ");
    for (int i = 0; i < strlen(s); i++)
    {
        d++;
        if (s[i] == ' ' && d > 3 && d < 7)
        {
            sr += d-1;
            n++;
            d = 0;
        }
    }
    printf("Prosečna dužina reci: %.3f.\n", sr / float(n));
}
```

3.(13b) Napisati funkciju **brisi** koja u zadatom nizu realnih brojeva  $a$  izbacuje sve elemente koji su van zadatog opsega ( $c, d$ ) (korisnik zadaje  $c$  i  $d$ ). Funkcija, dodatno, treba da ažurira dužinu niza nakon brisanja. Napisati glavni program koji testira ovu funkciju i ispisuje elementi niza nakon primene funkcije na niz koji zadaje korisnik.

```
#include <stdio.h>

void brisi(float a[], int &n, int c, int d)
//Brisanje elemenata iz a koji su van opsega (c,d).
{
    int j = 0;
    for (int i = 0; i < n; i++)
    {
        if (a[i] >= c && a[i] <= d)
            a[j++] = a[i];
    }
    n = j;
}

int main()
{
    int n;
    float niz[50], c, d;
    printf("Uneti duzinu niza: ");
    scanf_s("%d", &n);
    printf("Uneti granice c i d: ");
    scanf_s("%f%f", &c, &d);
    printf("Uneti elemente niza: ");
    for (int i = 0; i < n; i++) scanf_s("%f", &niz[i]);
    brisi(niz, n, c, d);
    printf("Niz nakon brisanja:\n");
    for (int i = 0; i < n; i++) printf("%f ", niz[i]);
}
```

4.(15b) Jedno autoprevozničko preduzeće raspolaže sa  $N$  vozila. Kreirati strukturu **autoprevoznik** sa dva elementa: broj pređenih kilometara (kao niz **bpk**[0.. $N-1$ ]) i potrošnja goriva svih vozila po kilometru (kao niz **pg**[0.. $N-1$ ]). Ukupnu cenu goriva, koja je plaćena za gorivo svih vozila, izračunava funkcija **totalno**, koja se poziva u glavnom programu. Obezbediti korisniku da u glavnom programu unosi  $N$ , pređene kilometre i potrošnju goriva za svako vozilo, kao i cenu goriva po litru.

```
#include <stdio.h>

struct autoprevoznik
{
    double bpk[20], pg[20];
};

double totalno(autoprevoznik a, int n, double c)
//Racunanje ukupne cene utrosenog goriva.
{
    double ukupno = 0;
    for (int i = 0; i < n; i++)
    {
        ukupno += a.bpk[i] * a.pg[i];
    }
    return ukupno * c;
}
```

```
int main()
{
    int N;
    autoprevoznik firma;
    double cena, jcena;
    printf("Uneti broj vozila: ");
    scanf_s("%d", &N);
    printf("Uneti predjene km po vozilu: ");
    for (int i = 0; i < N; i++) scanf_s("%lf", &firma.bpk[i]);
    printf("Uneti potrošnju goriva svakog vozila: ");
    for (int i = 0; i < N; i++) scanf_s("%lf", &firma.pg[i]);
    printf("Uneti cenu goriva po litru: ");
    scanf_s("%lf", &jcena);
    cena = totalno(firma, N, jcena);
    printf("Ukupna cena goriva za sva vozila: %.3lf.\n", cena);
}
```