



Универзитет „Св. Кирил и Методиј“ - Скопје  
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ  
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

## Структурно програмирање

Аудиториски вежби 6

# Содржина

1. Низи []	1
1.1. Декларација на низа	1
1.2. Пристап на елемент од низа	1
2. Задачи	1
2.1. Задача 1	1
2.2. Задача 2	2
2.3. Задача 3	3
2.4. Задача 4	3
2.5. Задача 5	4
2.6. Задача 6	5
2.7. Задача 7	6
3. Матрици (дво димензионални низи)	7
3.1. Задача 1	7
3.2. Задача 2	7
3.3. Задача 3	8
4. Изворен код од примери и задачи	10

## 1. Низи []

### 1.1. Декларација на низа

```
type variable_name[SIZE];  
  
int a[10];  
float x[99];  
char c[5];
```

### 1.2. Пристап на елемент од низа

```
array[element_index];  
  
int a[10];  
a[0] = 1; // доделување вредност 1 на првиот елемент  
printf("%d", a[9]); // печатење на вредноста на последниот елемент
```

## 2. Задачи

### 2.1. Задача 1

Да се напише програма која за две низи кои се внесуваат од тастатура ќе провери дали се еднакви или не. На екран да се испачати резултатот од споредбата. Максимална големина на низите е 100.

```
#include<stdio.h>
#define MAX 100
int main() {
    int n1, n2, element, i;
    int a[MAX], b[MAX];
    printf("Golemina na prvata niza: ");
    scanf("%d", &n1);
    printf("Golemina na vtorata niza: ");
    scanf("%d", &n2);
    if (n1 != n2)
        printf("Nizite ne se ednakvi\n");
    else {
        printf("Elementi na prvata niza: \n");
        for (i = 0; i < n1; ++i) {
            printf("a[%d] = ", i);
            scanf("%d", &a[i]);
        }
        printf("Elementi na vtorata niza: \n");
        for (i = 0; i < n2; ++i) {
            printf("b[%d] = ", i);
            scanf("%d", &b[i]);
        }
        // check if arrays are equal:
        for (i = 0; i < n1; ++i)
            if (a[i] != b[i])
                break;
        if (i == n1)
            printf("Nizite se ednakvi\n");
        else
            printf("Nizite ne se ednakvi\n");
    }
    return 0;
}
```

## 2.2. Задача 2

Да се напише програма која за низа чии елементи се внесуваат од тестатура, ќе го пресмета збирот на парните елементи, збирот на непарните елементи, како и односот помеѓу бројот на парни и непарни елементи. Резултатот да се испечати на екран.

### Пример

За низата: 3 2 7 6 2 5 1 На екран ќе се испечати:

```
Suma parni: 8
Suma neparni: 16
Odnos: 0.75
```

## Решение p6\_2.c

```
#include <stdio.h>
#define MAX 100
int main() {
    int i, n, a[MAX], brNep = 0, brPar = 0, sumNep = 0, sumPar = 0;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n; ++i) {
        if (a[i] % 2) {
            brNep++;
            sumNep += a[i];
        } else {
            brPar++;
            sumPar += a[i];
        }
    }
    printf("Suma parni: %d\nSuma neparni: %d\n", sumPar, sumNep);
    printf("Odnos: %.2f\n", (float)brPar / brNep);
    return 0;
}
```

## 2.3. Задача 3

Да се напише програма која ќе го пресмета скаларниот производ на два вектори со по  $n$  координати. Бројот на координати  $n$ , како и координатите на векторите се внесуваат од стандарден влез. Резултатот да се испечати на екран.

## Решение p6\_3.c

```
#include<stdio.h>
#define MAX 100
int main() {
    int a[MAX], b[MAX], n, i, scalar = 0;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n; ++i)
        scanf("%d", &b[i]);
    for (i = 0; i < n; ++i)
        scalar += a[i] * b[i];
    printf("Skalarniot proizvod e: %d\n", scalar);
    return 0;
}
```

## 2.4. Задача 4

Да се напише програма која ќе провери дали дадена низа од  $n$  елементи која се чита од стандарден влез е *строго растечка*, *строго опаѓачка* или ниту строго растечка ниту строго опаѓачка. Резултатот да се испечати на екран.

```
#include <stdio.h>
#define MAX 100
int main() {
    int n, element, a[MAX], i;
    short rastecka = 1, opagacka = 1;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n - 1; ++i) {
        if (a[i] >= a[i + 1]) {
            rastecka = 0;
            break;
        }
    }
    for (i = 0; i < n - 1; ++i) {
        if (a[i] <= a[i + 1]) {
            opagacka = 0;
            break;
        }
    }
    if (!opagacka && !rastecka)
        printf("Nizata ne e nitu rastechka nitu opagjachka\n");
    else if (opagacka)
        printf("Nizata e opagjachka\n");
    else if (rastecka)
        printf("Nizata e rastechka\n");
    return 0;
}
```

## 2.5. Задача 5

Да се напише програма за ротирање на елементите на една низа за едно место во десно. На крај, да се испечати на екран ротираната низа. Елементите од низата се читаат од стандарден влез.

```
/*
Да се напише програма за ротирање на елементите на една низа за едно место во десно.
*/
#include<stdio.h>
#define MAX 100
int main() {
    int n, i, temp;
    int a[MAX];
    printf("Golemina na niza: ");
    scanf("%d", &n);

    for(i = 0; i < n; ++i) {
        scanf("%d", &a[i]);
    }

    temp = a[n-1];
    for(i = n - 1; i > 0; i--) {
        a[i] = a[i-1];
    }
    a[0] = temp;

    for(i = 0; i < n; ++i) {
        printf("%d ", a[i]);
    }

    return 0;
}
```

## 2.6. Задача 6

Да се напише програма за ротирање на елементите на една низа за  $m$  места во десно. На крај, да се испечати на екран ротираната низа. Елементите од низата и бројот на ротирања се читаат од стандарден влез.

```
/*
Да се напише програма за ротирање на елементите на една низа за 'm' (се задава од
тастатура) места во десно.
*/
#include<stdio.h>
#define MAX 100
int main() {
    int n, i, j, temp, m;
    int a[MAX];
    printf("Golemina na niza: ");
    scanf("%d", &n);

    printf("Broj na rotiranja: ");
    scanf("%d", &m);

    for(i = 0; i < n; ++i) {
        scanf("%d", &a[i]);
    }

    for(i = 0; i < n; ++i) {
        printf("%d ", a[i]);
    }

    for(j = 0; j < m; j++) {
        temp = a[n-1];
        for(i = n - 1; i > 0; i--) {
            a[i] = a[i-1];
        }
        a[0] = temp;
    }

    printf("\n");
    for(i = 0; i < n; ++i) {
        printf("%d ", a[i]);
    }

    return 0;
}
```

## 2.7. Задача 7

Да се напише програма која што ќе ги избрише дупликатите од една низа. На крај, да се испечати на екран новодобиената низа. Елементите од низата се читаат од стандарден влез.



## Решение p6\_7.c

```

#include <stdio.h>
#define MAX 100
int main() {
    int a[MAX], n, i, j, k, izbrisani = 0;
    scanf("%d", &n);
    for (i = 0; i < n; ++i)
        scanf("%d", &a[i]);
    for (i = 0; i < n - izbrisani; ++i)
        for (j = i + 1; j < n - izbrisani; ++j)
            if (a[i] == a[j]) {
                for (k = j; k < n - 1 - izbrisani; ++k)
                    a[k] = a[k + 1];
                izbrisani++;
                --j;
            }
    n -= izbrisani;
    for (i = 0; i < n; ++i)
        printf("%d\t", a[i]);
    return 0;
}

```

## 3. Матрици (дво димензионални низи)

### 3.1. Задача 1

Да се напише програма која за матрица внесена од тастатура ќе ја пресмета разликата на збирот на елементите на непарните колони и збирот на елементите на парните редици. Матрицата не мора да биде квадратна.

## Решение p7\_1.c

```

#include<stdio.h>
#define MAX 100
int main() {
    int a[MAX][MAX], n, m, i, j, sumKol = 0, sumRed = 0;
    scanf("%d %d", &n, &m);
    for (i = 0; i < n; ++i)
        for (j = 0; j < m; ++j)
            scanf("%d", &a[i][j]);

    for (i = 0; i < n; ++i)
        for (j = 0; j < m; ++j) {
            if ((j + 1) % 2)
                sumKol += a[i][j];
            if (!((i + 1) % 2))
                sumRed += a[i][j];
        }
    printf("%d", sumKol - sumRed);
    return 0;
}

```

### 3.2. Задача 2

Да се напише програма која за матрица внесена од тастатура ќе ги замени

елементите од главната дијагонала со разликата од максималниот и минималниот елемент во матрицата. Резултантната матрица да се испечати на екран.

*Решение p7\_2.c*

```
#include <stdio.h>
#define MAX 100

int main() {
    int a[MAX][MAX];
    int n;
    scanf("%d", &n);
    int i, j;
    for (i = 0; i < n; ++i) {
        for (j = 0; j < n; ++j) {
            scanf("%d", &a[i][j]);
            if (i == 0 && j == 0) {
                min = max = a[i][j];
            } else if (a[i][j] > max) {
                max = a[i][j];
            } else if (a[i][j] < min) {
                min = a[i][j];
            }
        }
    }

    for (i = 0; i < n; ++i) {
        a[i][i] = max - min;
    }

    for (i = 0; i < n; ++i) {
        for (j = 0; j < n; ++j) {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

### 3.3. Задача 3

Да се напише програма која за квадратна матрица внесена од тастатура ќе испечати на екран дали таа е симетрична во однос на главната дијагонала.

```
#include <stdio.h>
#define MAX 100
int main () {
    int a[MAX][MAX], n, i, j, simetrichna = 1;
    printf("Vnesete dimenzija na kvadratna matrica: \n");
    scanf("%d", &n);
    printf("Vnesete gi elementite: \n");
    for (i = 0; i < n; ++i)
        for (j = 0; j < n; ++j)
            scanf ("%d", &a[i][j]);
    for (i = 0; i < n - 1; ++i) {
        for (j = i + 1; j < n; ++j)
            if (a[i][j] != a[j][i]) {
                simetrichna = 0;
                break;
            }
        if (!simetrichna) break;
    }
    if (simetrichna)
        printf("Matricata e SIMETRICHNA vo odnos na glavnata dijagonala\n");
    else
        printf("Matricata ne e SIMETRICHNA vo odnos na glavnata dijagonala\n");
    return 0;
}
```

## 4. Изворен код од примери и задачи

<https://github.com/finki-mk/SP/>

Source code ZIP