

КАТЕДРА: КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ
ДИСЦИПЛИНА: АЛГОРИТМИ СТРУКТУРИ ОТ ДАННИ
ДИСЦИПЛИНА: СИНТЕЗ И АНАЛИЗ НА АЛГОРИТМИ

ЛАБОРАТОРНО УПРАЖНЕНИЕ № 1

ТЕМА: Масиви. Обработка на масиви. (Преговор)

ЦЕЛ:

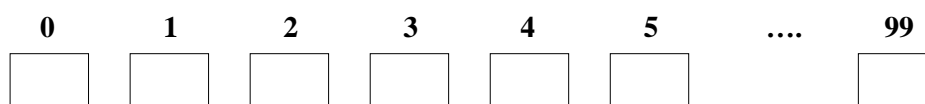
Целта на упражнението е студентите да преговорят и упражнят своите познания при работата с масиви. След упражнението студентите би следвало да нямат проблеми да боравят с тях.

I. ТЕОРЕТИЧНА ЧАСТ

1. Въведение.

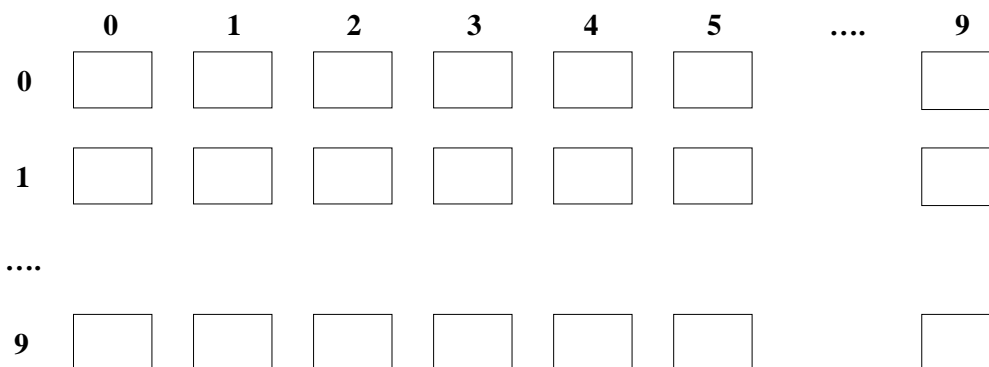
Масивът е структура от данни, състояща се от множество последователно наредени елементи от един и същи тип, достъпът до който се осъществява чрез името на масива, поредния номер на елемента, или т.нар. индекс.

Според размерността си се разделят на: едномерни и многомерни. Многомерните масиви се разделят на двумерни, тримерни и т.н. Едномерните масиви могат да се представят като редица от последователно наредени елементи (фиг.1). Двумерните се представят като матрици (фиг. 2).



Фиг.1. Структура на едномерен масив от 100 елемента

В паметта на компютъра многомерните масиви се представят като едномерни. Например, двумерен масив се нарежда последователно ред по ред. Двумерният масив се третира като едномерен масив, чиито елементи са едномерни масиви.



Фиг.2. Структура на двумерен масив от 10x10 елемента

Едномерни масиви се дефинират чрез следната декларация:

име_тип име_масив[размерност];

име_тип – определя типа на елементите на масива. Може да е някой от стандартните типове или дефиниран от програмиста;

име_масив – е идентификатор, определящ името на променливата от тип масив;

размерност – константен израз, който определя броя на елементите в масива и се нарича още граница.

Примери:

```
int massiv[10];      // дефиниране на едномерен масив massiv от 10 цели елемента
char ch[20];         // дефиниране на едномерен масив ch от 20 символа
float a[10], b[20];  // дефиниране на два едномерни, реални масива a и b от 10 и 20
                    // елемента, съответно
```

Достъпът до даден елемент на масива е с името на масива и индекса на елемента.

Пример:

```
a[2]=0;
b[18]=1;
ch[10]='y';
```

Особеност на език C, относно масивите е, че индексите на елементите започват от 0, т.е. ако даден масив е с N елемента, първият елемент е с номер 0, а последният е с номер N-1.

Дефинирането на многомерни масиви е чрез следната декларация:

име_тип име_масив[граница1][граница2].....;

Примери:

```
float ax[10][20];    // дефиниране на двумерен масив ax от 10x20 реални елемента
int bx [10][10][5];  // дефиниране на тримерен масив bx от 10x10x5 целочислени
                    // елемента
```

При дефинирането на масиви могат да бъдат зададени начални стойности на елементите, т.е. масивите да бъдат инициализирани. Общият вид на дефиницията е следния:

име_тип име_масив [размерност]={списък стойности};

Стойностите на елементите се разделят със запетаи.

Пример:

```
int array[3] = {1, 3, 5};
/* съответно стойностите на елементите са: array[0]=1; array[1]=3; array[2]=5 */
```

II. ПРАКТИЧЕСКА ЧАСТ

В практическата част са показани примери за използването на масиви като структури от данни.

ЗАДАЧА1: Да се намери позицията и стойността на минимален елемент в едномерен масив от 10 целочислени елемента.

КОД:

```

#include
main ()
{
    int mas[10];
    int i,j,a,k, min;
    cout<<"Insert the number of the massive=";
    cin>>j;

    for(a=0;a<j;a++)    cin>>mas[a];

    min=mas[0];          // min съдържа стойността на най-малкия елемент
    k=0;                 // k съдържа позицията на най-малкия елемент
    for (i=1; i<j;i++)
        if (mas[i]<min)
        {
            min=mas[i];
            k=i;
        }
    cout<<"min="<<min;
    cout<<"Poziciqta e"<<k;
    return 0;
}

```

ПОЯСНЕНИЕ

Задачата работи на следния принцип, взема първия елемент за минимален и го сравнява със всички останали. Ако открие друг елемент по малък от минималния, то ново открития елемент става новия минимум и за k се записва неговата позиция. Накрая програмата разпечатва стойността на минималния елемент и позицията му.

ЗАДАЧА2: Да се създаде програма която намира сумата на две целочислени матрици A и B, като резултата го записва в трета матрица.

КОД:

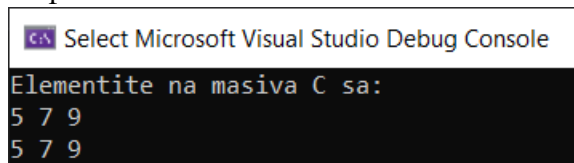
```

int main()
{
    int a[2][3] = { {1,2,3},{4,5,6} };
    int b[2][3] = { {4,5,6},{1,2,3} };
    int c[2][3];
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 3; j++)
            c[i][j] = a[i][j] + b[i][j];
    cout << "Elementite na masiva C sa: ";
    cout << endl;
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 3; j++)
            cout << c[i][j]<<" ";
        cout << endl;
    }
}

```

ПОЯСНЕНИЕ

Дефинирани са две матрици *a* и *b* с размерност 2x3. Първите два `for` цикъла пресмятат елементите на матрица *c*. Другите два `for` цикъла и извеждат елементите. Резултат от изпълнението е показан на фиг.3.



```
Select Microsoft Visual Studio Debug Console
Elementite na masiva C sa:
5 7 9
5 7 9
```

Фиг.3. Резултат от изпълнението на кода от зад. 2.

III. Задача за самостоятелна работа.

1. Тествайте задачите от практическата част.
2. Да се напише сорс код, с който се намира сумата и средно аритметичната стойност на елементите на едномерен масив от 10 реални числа с двойна точност.
3. Да се напише сорс код, с който се умножават две матрици. Стойностите на елементите на входните матрици да се въвеждат от клавиатурата.
4. Да се напише сорс код за въвеждане елементи на двумерен масив и намиране сумата на елементите във всеки ред.