

Технически Университет – филиал Пловдив
Факултет по Електроника и Автоматика

Протокол №:1

Тема: Работа с Многомерни масиви

Изготвил: Даниел Райчев Славчев

Факултетен номер: 382447

Специалност: КСТ

Дата: 23.11.2023г.

Група: 42б

1)Теория

В допълнение към едномерните масиви, в програмирането можем да работим и с многомерни масиви, които имат един или повече размерности. Всяка размерност се определя чрез собствена двойка от скоби. Например, изразът:

```
int ia[4][3];
```

дефинира двумерен масив. Първата размерност се отнася за редове, а втората - за колони. Този масив, ia, притежава четири реда по три елемента.

Многомерните масиви обикновено се наричат матрици. Можем също така да декларираме многомерни масиви с повече от две размерности. Например:

```
int arr3[5][10][7]; //декларира тримерен масив с размерности 5 x 10 x 7.
```

Размерностите на многомерните масиви предоставят удобен начин за индексирание, като те заемат последователни клетки в паметта, подобно на едномерните масиви.

Многомерните масиви могат да бъдат инициализирани. Например:

```
int ia[4][3] = { {0, 1, 2}, {3, 4, 5}, {6, 7, 8}, {9, 10, 11} };
```

Вътрешните скоби, указващи разделянето по редове, не са задължителни, но улесняват четимостта. Предходната инициализация може също да се напише по следния начин:

```
int ia[4][3] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 };
```

Когато вътрешните скоби липсват, резултатът ще бъде съвсем различен. Например:

```
int ia[4][3] = { 0, 1, 2, 3 };
```

Тук се инициализират първите три елемента на първия ред и първия елемент на втория. Останалите елементи получават стойност 0.

За индексирание на многомерни масиви се използват двойки скоби за всяка размерност. Например, следната двойка от два вложени цикъла for инициализира двумерен масив:

```
#define rowSize 4
```

```
#define colSize 3
```

```
void main() {
```

```
    int ia[rowSize][colSize];
```

```
    for (int i = 0; i < rowSize; ++i)
```

```
        for (int j = 0; j < colSize; ++j)
```

```
            ia[i][j] = i + j;
```

```
}
```

Този пример илюстрира начина, по който можем да използваме цикли за инициализация на елементите на двумерен масив.

2)Задачи

1) Напишете програма, която декларира тримерен масив с размерност 3x3x3 и го зарежда с числата от 1 до 27, като използва вложени цикли

```
#include <stdio.h>
void main()
{
    int arr[3][3][3];
    int a = 1;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            for (int k = 0; k < 3; k++)
            {
                arr[i][j][k] = a;
                a++;
            }
        }
    }

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            for (int k = 0; k < 3; k++)
            {
                printf("%d\n", arr[i][j][k]);
            }
        }
    }
}
```

C:\Users\c\Desktop\IntroProg\42b\protokol17>zad1.exe

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

2) Да се създаде програма, която обхожда двумерен масив с цели числа размер 5x3 и намира реда с най-голяма сума на елементите. Стойностите на елементите да се въведат от потребителя.

```

#include <stdio.h>
void main()
{
    int arr[5][3];
    int sumt;
    int sum;
    int num_of_row;
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("Input:");
            scanf("%d",&arr[i][j]);
        }
    }
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            printf("%d  ", arr[j][i]);
        }
        printf("\n");
    }
    for (int i = 0; i < 3; i++)
    {
        sum = 0;
        for (int j = 0; j < 5; j++)
        {
            sum += arr[j][i];
        }
        if (i == 0)
        {
            sumt = sum;
            num_of_row = i;
        }
        if (sum > sumt)
        {
            sumt = sum;
            num_of_row = i;
        }
    }
    printf("The row with the highest sum is %d and the sum is %d",num_of_row,sumt);
}

```

```

C:\Users\c\Desktop\IntroProg\42b\protokol7>zad2.exe
Input:1
Input:2
Input:3
Input:4
Input:5
Input:6
Input:7
Input:8
Input:9
Input:10
Input:11
Input:12
Input:13
Input:14
Input:15
1  4  7  10  13
2  5  8  11  14
3  6  9  12  15

```

3) Да се създаде програма, която обхожда двумерен масив с цели числасразмер 10x10 и намира индексите на елемента с най-малка стойност. Стойностите на елементите да се въведат от потребителя.

```

Input:-11111
Input:1
Input:2
Input:3
Input:4
Input:5
Input:56
Input:6
Input:7
Input:8
Input:9
Input:0
Input:11
Input:12
Input:122
Input:13
Input:14
Input:15
Input:16
Input:17
Input:18
Input:19
Input:20
Input:21
Input:22
The lowest number is -11111 [0][0]
C:\Users\c\Desktop\IntroProg\42b\protokol7>S

```

```

#include <stdio.h>
void main()
{
    int arr[5][5];
    int min;
    int indexi=0;
    int indexj=0;
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            printf("Input:");
            scanf("%d", &arr[i][j]);
        }
        min = arr[0][0];
        for (int i = 0; i < 5; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                if (arr[i][j] < min)
                {
                    min = arr[i][j];
                    indexi = i;
                    indexj = j;
                }
            }
        }
        printf("The lowest number is %d [%d][%d]", min, indexi, indexj);
    }
}

```

4) Даден е целочисления масив $A[7][2]$, чиито елементи имат стойност 1, 2 или 3. Да се изведе елемента, който се повтаря най-много. Ако иманяколко такива числа – да се изведе на екрана това, което е по-малкото стойност. Стойностите на елементите да се въведат от потребителя.

```

C:\Users\c\Desktop\IntroProg\42b\protokol7>zad4.exe
Input:1
Input:1
Input:1
Input:1
Input:2
Input:2
Input:2
Input:2
Input:2
Input:3
Input:3
Input:3
Input:3
Input:3
2 is the most common number in this array!(5 times)

```

```

#include <stdio.h>
void main()
{
    int arr[7][2];
    int one_count = 0;
    int two_count = 0;
    int three_count = 0;
    for (int i = 0; i < 7; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            printf("Input:");
            scanf("%d", &arr[i][j]);
            if (arr[i][j] == 1)
            {
                one_count++;
            }
            if (arr[i][j] == 2)
            {
                two_count++;
            }
            if (arr[i][j] == 3)
            {
                three_count++;
            }
        }
    }
    if ((one_count >= three_count) && (one_count >= two_count))
    {
        printf("1 is the most common number in this array!(%d times)", one_count);
    }
    else if ((two_count > one_count) && (two_count >= three_count))
    {
        printf("2 is the most common number in this array!(%d times)", two_count);
    }
    else
    {
        printf("3 is the most common number in this array!(%d times)", three_count);
    }
}

```

5) Даден е двумерния масив A[5][5], чиито елементи са цели числа. Намерете сбора на елементите по главния диагонал.

```
Input:1
Input:2
Input:3
Input:4
Input:5
Input:6
Input:7
Input:8
Input:9
Input:10
Input:11
Input:12
Input:13
Input:14
Input:15
Input:16
Input:17
Input:18
Input:19
Input:20
Input:21
Input:22
Input:23
Input:24
Input:25
Sum of the numbers on the main diagonal is 65
```

```
#include <stdio.h>
void main()
{
    int arr[5][5];
    int sum=0;
    for (int i = 0;i < 5;i++)
    {
        for (int j = 0; j < 5; j++)
        {
            printf("Input:");
            scanf("%d", &arr[i][j]);
            if (i == j)
            {
                sum += arr[i][j];
            }
        }
    }
    printf("Sum of the numbers on the main diagonal is %d", sum);
}
```


6) Дадени са целочислените масиви A[3][4] и B[3][4]. Да се създаде новмасив C[3][4], чиито елементи се образуват при умножението на елементот A и от B ($A[0][1] * B[0][1] = C[0][1]$).

```

Input in A:1
Input in B:1
Input in A:2
Input in B:2
Input in A:3
Input in B:3
Input in A:4
Input in B:4
Input in A:5
Input in B:5
Input in A:6
Input in B:6
Input in A:7
Input in B:7
Input in A:8
Input in B:8
Input in A:9
Input in B:9
Input in A:10
Input in B:10
Input in A:11
Input in B:11
Input in A:12
Input in B:12
1 25 81
4 36 100
9 49 121
16 64 144

1 #include <stdio.h>
2 void main()
3 {
4     int A[3][4];
5     int B[3][4];
6     int C[3][4];
7     for (int i = 0; i < 3; i++)
8     {
9         for (int j = 0; j < 4; j++)
10        {
11            printf("Input in A:");
12            scanf("%d", &A[i][j]);
13            printf("Input in B:");
14            scanf("%d", &B[i][j]);
15            C[i][j] = A[i][j] * B[i][j];
16        }
17    }
18    for (int i = 0; i < 4; i++)
19    {
20        for (int j = 0; j < 3; j++)
21        {
22            printf("%d ", C[j][i]);
23        }
24        printf("\n");
25    }
26 }

```

7) Напишете програма, инициализираща масив с размерност 10x3, така че първият елемент на всеки ред да съдържа число, вторият елемент да съдържа неговия квадрат, а третия - неговата трета степен.

```

D:\ТУ\ВП\ЛУ\protokol7>zad7.exe
Input a:1
Input a:2
Input a:3
Input a:4
Input a:5
Input a:6
Input a:7
Input a:8
Input a:9
Input a:10
1 2 3 4 5 6 7 8 9 10
1 4 9 16 25 36 49 64 81 100
1 8 27 64 125 216 343 512 729 1000

```

```

1  #include <stdio.h>
2  void main()
3  {
4      int arr[10][3];
5      for (int i = 0; i < 10; i++)
6      {
7          for (int j = 0; j < 3; j++)
8          {
9              if (j == 0)
10             {
11                 printf("Input a:");
12                 scanf("%d", &arr[i][j]);
13             }
14             else
15             {
16                 arr[i][j] = arr[i][j - 1] * arr[i][0];
17             }
18         }
19     }
20     for (int i = 0; i < 3; i++)
21     {
22         for (int j = 0; j < 10; j++)
23         {
24             printf("%d  ", arr[j][i]);
25         }
26         printf("\n");
27     }
28 }

```

8) Даден е двумерния масив `int Rotate[a][b]`. Да се създадат функции, които:

- извежда стойностите на масива по колони и редове
- разменя елементите, така че редовете да станат колони, а колоните редове.

```

Input a:1
Input a:2
Input a:3
Input a:4
Input a:5
Input a:6
Input a:7
Input a:8
Input a:9
Input a:10
1 2 3 4 5 6 7 8 9 10
1 4 9 16 25 36 49 64 81 100
1 8 27 64 125 216 343 512 729 1000
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000

```

```
1  #include <stdio.h>
2  void main()
3  {
4      int arr[10][3];
5      for (int i = 0; i < 10; i++)
6      {
7          for (int j = 0; j < 3; j++)
8          {
9              if (j == 0)
10             {
11                 printf("Input a:");
12                 scanf("%d", &arr[i][j]);
13             }
14             else
15             {
16                 arr[i][j] = arr[i][j - 1] * arr[i][0];
17             }
18         }
19     }
20     for (int i = 0; i < 3; i++)
21     {
22         for (int j = 0; j < 10; j++)
23         {
24             printf("%d ", arr[j][i]);
25         }
26         printf("\n");
27     }
28     printf("\n");
29     for (int i = 0; i < 10; i++)
30     {
31         for (int j = 0; j < 3; j++)
32         {
33             printf("%d ", arr[i][j]);
34         }
35         printf("\n");
36     }
37 }
38
```