



جامعة حماه  
المعهد التقني للحاسوب  
السنة الأولى

محاضرة ٤

برمجة ٢

عملي

قسم البرمجيات

إعداد:

م. أريج فياض

م. رفا البنات

## نتيجة:

أنواع الوسطاء أي البارامترات التي يمكن للتابع (الطرق) أن تأخذها هي :

١ - وسيط بالقيمة : عندما نمرر نسخة عن الوسيط للتابع وأي تغيير على هذه النسخة ضمن التابع لن يؤثر على النسخة الأصلية من هذا الوسيط الممرر.

٢- وسيط بالمرجع :عندما نرسل مؤشر على نفس عنوان الوسيط الأصلي ليأخذه التابع فأي تغيير عليه ضمن التابع سيؤثر على الوسيط حتماً.

٣- وسيط بالخرج : عندما نريد من التابع أن يعيد أكثر من قيمة.

٤ - مصفوفة وسطاء :عندما نريد أن نمرر للتابع أكثر من قيمة فنمرر له مصفوفة فيها تلك الوسطاء التي تكون من نفس النوع ويقوم التابع بالعمل على عناصر تلك المصفوفة.

• استخدام المصفوفات كوسطاء للتابع:

عند التصريح عن التابع الذي يأخذ وسيط من نوع مصفوفة نقوم بما يلي:

( التصريح عن حجم المصفوفة ,التصريح عن المصفوفة) اسم التابع نوع القيمة المعادة static {

أكواد التابع

}

ثم نقوم باستدعاء التابع ضمن ال main .

أمثلة:

- اكتب تابع يقوم بإدخال عناصر مصفوفة أحادية ثم طباعتها بشكل أفقي.

```
static void read(int[] array, int n)
{
    for (int i = 0; i < n; i++)
        array[i] = int.Parse(Console.ReadLine());
}
static void Main(string[] args)
{
    Console.WriteLine("enter the number of array elements:");
    int m = int.Parse(Console.ReadLine());
    Console.WriteLine("enter the elements:");
    int[] array = new int[m]; // المصفوفة وبنائها قبل تمريرها للتابع
    read(array, m); // استدعاء التابع
    Console.WriteLine("the array elements are:");
    for (int i = 0; i < m; i++)
    { Console.Write(array[i] + " "); }
    Console.ReadKey(); }
```

```

enter the number of array elements:
5
enter the elements:
48
62
79
12
58
the array elements are:
48 62 79 12 58

```

كما يمكن أن نمرر بارامترت أخرى نتعامل مع عناصر المصفوفة .

- اكتب تابع للبحث عن عنصر داخل مصفوفة أحادية.

```

static bool search(int[] x, int n, int num)
{
    int box = 0;
    for (int i = 0; i < n; i++)

        if (x[i] == num)
        { box = 1; }
    if (box == 1)
    { return true; }
    else
        return false;
}

static void Main(string[] args)
{
    Console.WriteLine("enter the element of array:");
    int[] x = new int[4];
    for (int k = 0; k < 4; k++)
        x[k] = int.Parse(Console.ReadLine());
    Console.WriteLine("enter the number for search:");
    int number = int.Parse(Console.ReadLine());
    bool result = search(x, 4, number);
    if (result == true)
    { Console.WriteLine("the number is exsit"); }
    else
    { Console.WriteLine("the number is not exsit"); }
    Console.ReadKey();
}

```

```

enter the element of array:
45
68
25
85
enter the number for search:
25
the number is exsit

```

في حال أردنا تمرير مصفوفة ثنائية:

{ static (عدد الأعمدة، عدد الأسطر، نوع المصفوفة) اسم التابع نوع القيمة المعادة

جسم التابع

}

- اكتب تابع لإدخال عناصر مصفوفة ثنائية ثم تابع يقوم بطباعة عناصرها.

```
{static void read(int[,] array , int row , int column)
{for(int i=0; i<row;i++)
{
    for (int j = 0; j < column; j++)
        array[i, j] = int.Parse(Console.ReadLine());
}
} //end of read method
static void write(int[,] array, int row, int column)
{
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < column; j++)
        { Console.Write(array[i, j] + " "); }
        Console.WriteLine();
    }
} //end of write method
static void Main(string[] args)
{
    Console.WriteLine("enter the number of rows:");
    int n = int.Parse(Console.ReadLine());
    Console.WriteLine("enter the number of columns:");
    int m = int.Parse(Console.ReadLine());
    Console.WriteLine("enter the elements of array:");
    int[,] x = new int[n, m];
    read(x, n, m);
    Console.WriteLine("the array is:");
    write(x, n, m);
    Console.ReadKey();
}
```

```
enter the number of rows:
3
enter the number of columns:
4
enter the elements of array:
2
8
6
1
8
6
7
4
3
6
9
8
the array is:
2 8 6 1
8 6 7 4
3 6 9 8
```

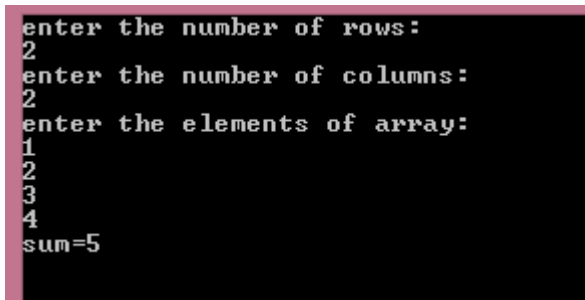
- اكتب تابع يقوم بجمع عناصر القطر الرئيسي لمصفوفة ثنائية.

```
static int sum(int[,] array, int n, int m)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (i == j)
            { s = s + array[i, j]; }
        }
    }
    return s;
}

static void Main(string[] args)
{
    Console.WriteLine("enter the number of rows:");
    int x = int.Parse(Console.ReadLine());
    Console.WriteLine("enter the number of columns:");
    int y = int.Parse(Console.ReadLine());
    Console.WriteLine("enter the elements of array:");
    int[,] array = new int[x, y];
    for (int i = 0; i < x; i++)
    {
        for (int j = 0; j < y; j++)
        {
            array[i, j] = int.Parse(Console.ReadLine());
        }
    }

    int z=sum(array, x, y);
    Console.WriteLine("sum=" + z);

    Console.ReadKey();
}
```



```
enter the number of rows:
2
enter the number of columns:
2
enter the elements of array:
1
2
3
4
sum=5
```

- اكتب تابع يقوم بحساب المتوسط الحسابي لعناصر مصفوفة أحادية بالاعتماد على تابع يقوم بجمع عناصرها.

```
static int sum(int[] x, int n)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    { s = s + x[i]; }

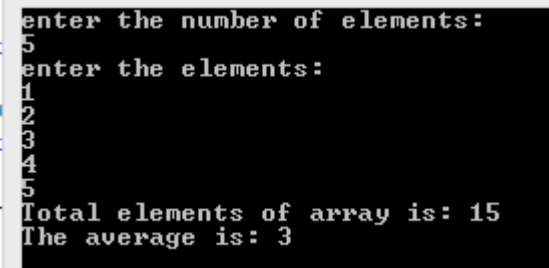
    return s;
}
static double avgerage(int[] y, int m)
{
    double avg = (double)sum(y, m) / m;
    return avg;
}
static void Main(string[] args)
{
    Console.WriteLine("enter the number of elements:");
    int n = int.Parse(Console.ReadLine());

    {
        Console.WriteLine("enter the elements:");
        int[] array = new int[n];

        for (int i = 0; i < n; i++)
        {
            array[i] = int.Parse(Console.ReadLine());
            Console.WriteLine("Total elements of array is: " + sum(array, n));

            Console.WriteLine("The average is: " + avgerage(array, n));

            Console.ReadKey();
        }
    }
}
```



```
enter the number of elements:
5
enter the elements:
1
2
3
4
5
Total elements of array is: 15
The average is: 3
```

- اكتب تابع يقوم بجمع عناصر السطر الأول من مصفوفة ثنائية .

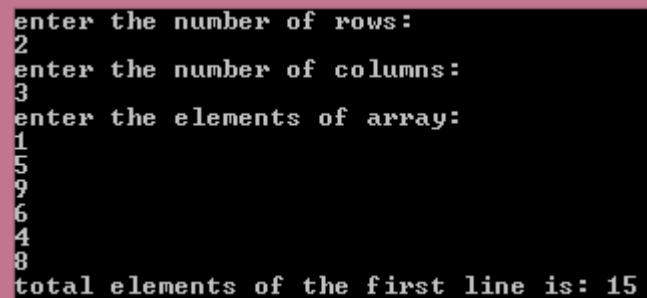
```
static int sum1(int[,] x, int n, int m)
{
    int s = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (i == 0)
            { s = s + x[i, j]; }
        }
    }
    return s;
}

static void Main(string[] args)
{
    Console.WriteLine("enter the number of rows:");
    int n = int.Parse(Console.ReadLine());

    Console.WriteLine("enter the number of columns:");
    int m = int.Parse(Console.ReadLine());

    {
        Console.WriteLine("enter the elements of array:");
        int[,] array = new int[n, m];
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                array[i, j] = int.Parse(Console.ReadLine());
            }
        }
        Console.WriteLine("total elements of the first line is: " + sum1(array, n, m));

        Console.ReadKey();
    }
}
```



```
enter the number of rows:
2
enter the number of columns:
3
enter the elements of array:
1
5
9
6
4
8
total elements of the first line is: 15
```

- اكتب تابع يقوم بإجراء عمليات حسابية على عددين حيث يقوم المستخدم بإدخال العددين والعلية.

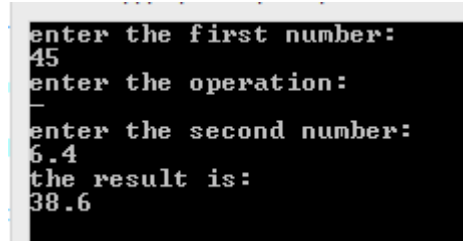
```
static double F(double x, char z, double y)
{
    switch (z)
    {
        case '+': { return (x + y); break; }
        case '-': { return (x - y); break; }
        case '*': { return (x * y); break; }
        case '/': { return (x / y); break; }
        default: return 0;
    }
}
static void Main(string[] args)
{
    Console.WriteLine("enter the first number:");
    double a = double.Parse(Console.ReadLine());

    Console.WriteLine("enter the operation:");
    char z = char.Parse(Console.ReadLine());

    Console.WriteLine("enter the second number:");
    double b = double.Parse(Console.ReadLine());

    Console.WriteLine("the result is:");
    Console.WriteLine(F(a,z,b));

    Console.ReadKey();
}
```



```
enter the first number:
45
enter the operation:
+
enter the second number:
6.4
the result is:
38.6
```

#### ملاحظات:

- \*١ يمكن أن يكون التابع بلا وسطاء .
- \*٢ ويجب أن يكون التابع الذي يستدعي توابعنا المنشأة من نوع static.
- \*٣ عند التصريح عن التابع المتغيرات التي توضع بين قوسين تسمى parametr

مثال: sum(int[,] array, int n, int m)

وعند استدعاء التابع وتمرير القيم الفعلية نسميهم argument

مثال: sum(array, x, y);

التصريح عن المصفوفة وبنائها باستخدام new قبل استدعاء التابع و تمريرها إليه.