

Trường ĐH CT TP. HCM Khoa: CNTT Bộ môn: CNPM Môn: Lập trình .NET	<h2 style="text-align: center;">BÀI 2</h2> <h1 style="text-align: center;">MẢNG, MẢNG CHUỖI</h1>	
---	--	--

A. MỤC TIÊU:

- Sử dụng cấu trúc mảng một chiều thực hiện các bài toán cơ bản.
- Biết cách xử lý dùng mảng một chiều trong C#.
- Sử dụng cấu trúc mảng hai chiều thực hiện các bài toán cơ bản.
- Biết cách xử lý dùng mảng hai chiều trong C#.

B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

a. Mảng một chiều:

Cú pháp khai báo:

```
<Kiểu dữ liệu>[ ] <Tên_mảng>;
```

Ví dụ: khai báo mảng số nguyên A (không xác định số phần tử tối đa)

```
int[] A;
```

Khởi tạo mảng A bằng từ khóa 'new'. Cú pháp là tiết kiệm vùng nhớ vì chỉ khai báo đủ nhu cầu sử dụng.

```
A = new int[n];
```

b. Mảng hai chiều:

1[0,0]	2[0,1]	4[0,2]	9[0,3]	3[0,4]	...[0,...]	m (cột) [0,m]
6[0,1]	4	3	6
...
n (dòng) [0,n]

Cú pháp khai báo:

```
<Kiểu dữ liệu>[ ][ ] <Tên_mảng>;
```

Ví dụ: khai báo mảng 2 chiều số nguyên A (không xác định số phần tử tối đa)

```
int[][] A;
```

Khởi tạo mảng A bằng từ khóa ‘new’ với n dòng và m là cột:

```
A = new int[n][m];
```

c. Chuỗi

Khái niệm: *Chuỗi là mảng một chiều chứa các ký tự.*

Ví dụ:

“N”	“g”	“u”	“y”	“e”	“n”
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Cú pháp khai báo:

```
string s;
```

Một số phương thức thao tác với chuỗi:

Phương thức	Ý nghĩa
Length	Chiều dài chuỗi
IndexOf (s1)	Vị trí xuất hiện đầu tiên của chuỗi s1 trong chuỗi ban đầu
LastIndexOf (s1)	Giống IndexOf, nhưng tìm lần xuất hiện cuối cùng
Replace (s1, s2)	Thay thế chuỗi s1 thành s2 trong chuỗi ban đầu
Split (<ký tự ngắt>)	Tách chuỗi thành mảng chuỗi bởi <ký tự ngắt>
Substring (vt, l)	Trả về chuỗi con bắt đầu ở một vị trí vt và chiều dài l.
ToLower ()	Chuyển chuỗi thành chữ thường
ToUpper ()	Chuyển chuỗi thành chữ in
Trim ()	Xóa khoảng trắng ở đầu và cuối chuỗi

1.2 Giới thiệu bài tập mẫu

Bài 1: Viết phương thức nhập mảng a gồm n phần tử, xuất mảng a.

Công việc 1: Phương thức xử lý nhập vào mảng a.

Phương thức có thể có hoặc không có giá trị trả về, tham số truyền vào là mảng 1 chiều a và số phần tử mảng.

Bước 1: Cấu trúc phương thức nhập mảng 1 chiều A:

```
static int[] NhapMang(int[] A, int n) {  
    // Xử lý  
    return A; }
```

Bước 2: Viết code xử lý công việc nhập mảng

```
static int[] NhapMang(int[] A, int n)
{
    // Xử lý
    for (int i = 0; i < n; i++)
    {
        Console.Write("Mời nhập phần tử A[{0}]=", i);
        A[i] = int.Parse(Console.ReadLine());
    }
    return A;
}
```

Công việc 2: Phương thức xử lý xuất mảng A.

Bước 1: Cấu trúc phương thức xuất mảng 1 chiều A

```
static void XuatMang(int[] A)
{
    // Xử lý
}
```

Bước 2: Viết code xử lý công việc xuất mảng

```
static void XuatMang(int[] A)
{
    // Xử lý
    for (int i = 0; i < A.Length; i++)
    {
        Console.Write("\t {0}", A[i]);
    }
}
```

Viết phương thức tổng quát gọi kiểm tra kết quả 2 phương thức trên:

```
public static void Mang1Chieu()
{
    // Khai báo mảng A
    int[] A;
    Console.Write("Mời bạn nhập vào số phần tử: ");
    // Nhập n
    int n = int.Parse(Console.ReadLine());
    // Khởi tạo mảng A đúng n phần tử
}
```

```

    A = new int[n];
    //Gọi phương thức nhập mảng
    A = NhapMang(A,n);
    //Gọi phương thức xuất mảng
    XuatMang(A);
}

```

✚ Một số kỹ thuật xử lý mảng 1 chiều:

- **Sắp xếp:**

```

static void SapXep(int[] A)
{
    // Sắp xếp tăng
    Array.Sort(A);
}

```

- **Đảo mảng:** `Array.Reverse();`
- **Thay đổi kích thước mảng:** `Array.Resize();`
- **Chuyển mảng từ kiểu string sang mảng kiểu int:**

```

string[] B= new[] { "1", "2" };
int[] intArray = Array.ConvertAll(B, int.Parse);

```

Ngoài ra mảng còn hỗ trợ trực tiếp các phương thức static thực hiện các công việc:

- Tính tổng các phần tử trong mảng: `A.Sum();`
- Tính trung bình các phần tử trong mảng: `A.Average();`
- Đếm số phần tử có trong mảng: `A.Count();`
- Tìm phần tử nhỏ nhất trong mảng: `A.Min();`
- Tìm phần tử lớn nhất trong mảng: `A.Max();`
- Tính tổng phần tử trong mảng theo điều kiện:

```

A.Where(t => t % 2 == 0).Sum();

```

Bài 2: Viết phương thức nhập mảng 2 chiều a gồm n phần tử, xuất mảng 2 chiều.

Công việc 1: Phương thức xử lý nhập vào mảng 2 chiều a.

Phương thức có thể có hoặc không có giá trị trả về, tham số truyền vào là mảng 2 chiều a và số phần tử mảng.

Bước 1: Cấu trúc phương thức nhập mảng 2 chiều

```

static void NhapMang2Chieu(int[][] A, int n, int m) {
    // Xử lý
}

```

Bước 2: Viết code xử lý công việc nhập mảng

```
static void NhapMang2Chieu(int[][] A, int n, int m)
{
    // Xử lý
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            Console.Write("Nhập phần tử A[{0}][{1}]= ", i, j);
            A[i][j] = int.Parse(Console.ReadLine());
        }
    }
}
```

Công việc 2: Phương thức xử lý xuất mảng a.

Bước 1: Cấu trúc phương thức xuất mảng

```
static void XuatMang2Chieu(int[][] A)
{
    // Xử lý
}
```

Bước 2: Viết code xử lý công việc xuất mảng

```
static void XuatMang2Chieu(int[][] A)
{
    // Xử lý
    for (int i = 0; i < A.GetLength(0); i++)
    {
        for (int j = 0; j < A.GetLength(1); j++)
        {
            Console.Write("{0} ", A[i][j]);
        }
        Console.WriteLine();
    }
}
```

Chúng ta có thể thay thế n, m bằng phương thức GetLength().

Viết phương thức tổng quát gọi kiểm tra kết quả 2 phương thức trên

```
static void Mang2Chieu()
{
    int n, m;
    int[][] A;
    Console.WriteLine("Mời bạn nhập số dòng: ");
    n = int.Parse(Console.ReadLine());
    Console.WriteLine("Mời bạn nhập số cột: ");
    m = int.Parse(Console.ReadLine());
    // Khởi tạo
    A = new int[n][];
    for (int i = 0; i < n; i++)
    {
        A[i] = new int[m];
    }
    NhapMang2Chieu(A, n, m);
    XuatMang2Chieu(A);
}
```

Một số kỹ thuật xử lý mảng 2 chiều:

🔍 **Lưu ý:** Tận dụng các kỹ thuật của mảng 1 chiều. Vì mảng 2 chiều là sự kết hợp của nhiều mảng 1 chiều.

Ví dụ: Tổng các số dương trong mảng 2 chiều:

```
static int TongDuong2Chieu(int[][] A)
{
    int tong = 0;
    for (int i = 0; i < A.GetLength(0); i++)
    {
        tong = A[i].Where(t => t > 0).Sum();
    }
    return tong;
}
```

Bài 3: Nhập chuỗi S. Kiểm tra S có là chuỗi đối xứng không ?

Bước 1: Nhập chuỗi S.

```
static void Main(string[] args)
{
```

```
string S;  
S = Console.ReadLine();  
Console.ReadLine();  
}
```

Bước 2: Chuyển chuỗi S thành mảng chuỗi.

```
static void Main(string[] args)  
{  
    string S;  
    S = Console.ReadLine();  
    char[] mangChuoi = S.ToCharArray();  
    Console.ReadLine();  
}
```

Bước 3: Viết phương thức kiểm tra đối xứng

```
static bool ChuoiDoiXung(char[] A)  
{  
    for (int i = 0; i < A.Length; i++)  
    {  
        if (A[i] != A[A.Length - 1 - i])  
            return false;  
    }  
    return true;  
}
```

Bước 4: Hàm Main gọi thực thi các phương thức trên

```
static void Main(string[] args)  
{  
    string S;  
    S = Console.ReadLine();  
    char[] mangChuoi = S.ToCharArray();  
    if (ChuoiDoiXung(mangChuoi))  
    {  
        Console.Write("Chuoi doi xung");  
    }  
    else  
    {  
        Console.Write("Chuoi khong doi xung");  
    }  
}
```

```
}  
    Console.ReadLine();  
}
```

2. Bài tập tại lớp

Bài 1. Mảng một chiều

Cho mảng 1 chiều chứa các số nguyên. Viết phương thức thực hiện:

- Nhập mảng a gồm n phần tử. Xuất mảng a.
- Liệt kê các phần tử lẻ ở vị trí chẵn.
- Liệt kê các số nguyên tố trong mảng.
- Tìm phần tử âm đầu tiên trong mảng.
- Tìm max, min của dãy.
- Tính tổng các phần tử trong mảng.

Bài 2. Mảng hai chiều

Cho một ma trận nguyên kích thước $m \times n$. Viết phương thức thực hiện:

- Nhập các giá trị cho mảng 2 chiều, sau đó xuất mảng vừa nhập.
- Tổng tất cả các phần tử dương của ma trận.
- Tổng các phần tử trên đường chéo chính.
- Tổng các phần tử trong tam giác trên.
- Tổng tất cả các phần tử chẵn của ma trận.
- Tính tổng các phần tử ở dòng thứ i.

Bài 3. Nhập chuỗi s (là họ tên). Viết phương thức thực hiện các yêu cầu sau:

- S có phải là chuỗi đối xứng không?
- Đổi ký tự đầu từ thành chữ hoa
- Đổi ký tự thường thành chuỗi hoa và ngược lại
- Có bao nhiêu nguyên âm, phụ âm, khoảng trắng trong chuỗi s.

3. Bài tập nâng cao

Bài 1. Thực hiện phép cộng 2 đa thức trên mảng một chiều. Với đa thức cho trước như sau: DaThuc {int Bac, int ChiSoDaThuc[]};

Bài 2. Cho một ma trận nguyên kích thước $m \times n$. Viết phương thức tìm:

- Số nhỏ nhất, lớn nhất (kèm chỉ số) của ma trận.
- Số nhỏ nhất, lớn nhất (kèm chỉ số) của từng hàng của ma trận.
- Số nhỏ nhất, lớn nhất (kèm chỉ số) của đường chéo chính của ma trận.
- Viết phương thức tìm giá trị phần tử xuất hiện nhiều nhất trong dòng i, nếu có nhiều phần tử có số lần xuất hiện bằng nhau thì xuất phần tử đầu tiên.

Bài 3. Nhập chuỗi s (là họ tên). Viết phương thức thực hiện các yêu cầu sau:

- Xóa khoảng trắng thừa trong chuỗi s.
- Có bao nhiêu từ trong chuỗi s
- Lấy ra họ
- Lấy ra tên
- Lấy ra họ lót (nếu có)

4. Bài tập về nhà

Bài 1. Viết phương thức tính tổng các phần tử cực đại trong mảng các số nguyên (phần tử cực đại là phần tử lớn hơn các phần tử xung quanh nó).

Ví dụ: 1 **5** 2 **6** 3 **5** 1 **8** 6

Bài 2. Cho ma trận vuông cấp n chứa các số nguyên. Viết phương thức thực hiện các yêu cầu sau:

- Liệt đường chéo chính, phụ.
- Liệt kê tam giác trên, dưới .
- Tổng đường chéo chính, phụ.
- Tổng tam giác trên, dưới .

-----Hết-----