

## Bài 3

### Bài tập mảng số nguyên

Email:thongt@dlu.edu.vn

#### Mục tiêu bài thực hành:

1. Khai báo mảng 1 chiều
2. Nhập dữ liệu cho mảng 1 chiều
3. Xuất mảng 1 chiều
4. Các thao tác trên mảng 1 chiều
5. Viết, và gọi hàm
6. Xác định phạm vi của biến

Bước 1: Tạo một Solution kiểu Console với tên Mang1ChieuSoNguyen

Bước 2: Khai báo mảng 1 chiều và biến lưu chiều dài của mảng

```
namespace Mang1ChieuSoNguyen
{
    0 references
    class Program
    {
        //Khai báo mảng 1 chiều và kích thước của mảng, khai báo biến ở đây
        //se có phạm vi sử dụng trong class Program
        static int[] a = new int[100]; //khai báo mảng 1 chiều a số nguyên với kích thước tối đa là 100 phần tử
        static int length = 0; //chiều dài của mảng
        0 references
        static void Main(string[] args)
        {
            ...
        }
    }
}
```

Bước 3: Viết hàm Nhập để nhập dữ liệu cho mảng

```

class Program
{
    //Khai bao mang 1 chieu va kích thước của mang, khai báo biến ở đây
    //se có phạm vi sử dụng trong class Program
    static int[] a = new int[100]; //khai báo mang 1 chiều a số nguyên với kích thước tối đa là 100 phần tử
    static int length = 0; //chiều dài của mang
    0 references
    static void Main(string[] args)
    {
    }
    0 references
    static void Nhap()
    {
        Console.WriteLine("Nhập vào số phần tử của mảng ");
        length = int.Parse(Console.ReadLine());
        for (int i = 0; i < length; i++)
        {
            Console.WriteLine(" a[{0}] = ", i);
            a[i] = int.Parse(Console.ReadLine());
        }
    }
}

```

Bước 4: Gọi hàm Nhap trong hàm Main để kiểm tra. Nhấn F5 để chạy chương trình.

```

class Program
{
    //Khai bao mang 1 chieu va kích thước của mang, khai báo biến ở đây
    //se có phạm vi sử dụng trong class Program
    static int[] a = new int[100]; //khai báo mang 1 chiều a số nguyên với kích thước tối đa là 100 phần tử
    static int length = 0; //chiều dài của mang
    0 references
    static void Main(string[] args)
    {
        Nhap();
        Console.ReadKey();
    }
    1 reference
    static void Nhap()
    {
        Console.WriteLine("Nhập vào số phần tử của mảng ");
        length = int.Parse(Console.ReadLine());
        for (int i = 0; i < length; i++)
        {
            Console.WriteLine(" a[{0}] = ", i);
            a[i] = int.Parse(Console.ReadLine());
        }
    }
}

```

### Chú ý:

Khi viết một hàm bất kỳ, thì thực hiện các bước giống như hàm Nhap vừa rồi.

1. Viết hàm, khi viết hàm chú ý: tên hàm; kiểu trả về; tham số của hàm.
2. Gọi hàm trong hàm Main để kiểm tra kết quả.

Bước 5: Thực hiện tương tự đề viết và chạy kiểm tra các hàm sau:

```
static void Main(string[] args)
{
    NhapNgauNghien();
    Xuat();
    //Console.WriteLine("Tong cac phan tu la " + TinhTong());
    int y = 0;
    Console.Write("Nhap vao phan tu can xoa");
    y = int.Parse(Console.ReadLine());
    //int vt= TimViTriDauTien(y);
    //Console.WriteLine("Vi tri cua {0} la {1} ",y,vt);
    XoaPhanTuDauTien(y);
    Xuat();
    Console.ReadKey();
}
1 reference
static void XoaPhanTuDauTien(int x)
{
    XoaPhanTuTaiViTri(TimViTriDauTien(x));
}
1 reference
static int TimViTriDauTien(int x)
{
    for (int i = 0; i < length; i++)
    {
        if (a[i] == x) return i;
    }
    return -1;
}
```

```

static void XoaPhanTuTaiViTri(int vt)
{
    for (int i = vt; i < length - 1; i++)
    {
        a[i] = a[i + 1];
    }
    //cho nay van con sai
    length--;
}
0 references
static int TimViTriCuoiCung(int x)
{
    for (int i = length - 1; i >= 0; i--)
    {
        if (a[i] == x) return i;
    }
    return -1;
}
1 reference
static void NhapNgauNghien()
{
    Console.Write("Nhap vao so phan tu cua mang ");
    length = int.Parse(Console.ReadLine());
    Random r = new Random();
    for (int i = 0; i < length; i++)
    {
        a[i] = r.Next(100);
    }
}

```

```

static int TinhTongCacSoNguyen()
{
    int tong = 0;
    for (int i = 0; i < length; i++)
    {
        tong += a[i];
    }
    return tong;
}

2 references
static void Xuat()
{
    Console.WriteLine("Mang vua nhap la ");
    for (int i = 0; i < length; i++)
    {
        Console.Write("\t " + a[i]);
    }
}

```

Bước 6: Thực hiện các yêu cầu sau đối với mảng 1 chiều:

**Chú ý:**

Không cần thực hiện theo thứ tự, chọn những hàm nào hiểu và dễ để viết trước. Nếu không hiểu các yêu cầu thì hỏi trên diễn đàn

```

0 references
static int TimPhanTuLonNhat()...
0 references
static int TimPhanTuNhoNhat()...
//Tra ve mang so nguyen va tham so lengthkq chua chieu dai mang nay
0 references
static int[] TimTatCaCacSoAm(ref int lengthkq)...
0 references
static int[] TimTatCaCacSoDuong(ref int lengthkq)...
0 references
static int[] TimTatCaCacChan(ref int lengthkq)...
0 references
static int[] TimTatCaCacSoLe(ref int lengthkq)...
0 references
static int[] TimTatCaCacSoNguyenTo(ref int lengthkq)...
0 references
static int[] TimPhanTuXuatHienNhiềuNhat(ref int lengthkq)...
0 references
static int[] TimPhanTuXuatHienItNhat(ref int lengthkq)...
//Tim tat ca cac phan tu lon hon so x nao do
0 references
static int[] TimTatCaPhanTuLonHon(ref int lengthkq, int x)...
0 references
static int[] TimTatCaPhanTuNhoHon(ref int lengthkq, int x)...
//Tu vi tri lay soLuongPhanTu
0 references
static int[] TimTatCaPhanTuViTri(ref int lengthkq, int vt, int soLuong)...
0 references

```

```

0 references
static bool ThemPhanTuTaiViTri(int x, int vt)...
0 references
static bool ThemPhanTuDauDanhSach(int x)...
0 references
static bool ThemPhanTuCuoiDanhSach(int x)...
//Them tat ca mang b vao mang a tai vi tri vt
0 references
static bool ThemMotMangVaoDanhSachTaiViTri(int[] b, int lengthb, int vt)...
0 references
static bool ThemMotMangVaoDauDanhSach(int[] b, int lengthb)...
0 references
static bool ThemMotMangVaoCuoiDanhSach(int[] b, int lengthb)...
0 references
static bool XoaPhanTuDau(int vt)...
0 references
static bool XoaPhanTuCuoi(int vt)...
1 reference
static bool XoaPhanTuDauTien(int x)...
1 reference
static bool XoaPhanTuTaiViTri(int vt)...
0 references
static bool XoaPhanTuCoTrongMang(int[] b, int lengthb)...
0 references
static bool XoaTatCaPhanTu()...
0 references
static bool XoaTatCaSoAm()...
0 references
static bool XoaTatCaSoDuong()...
0 references
static bool XoaTatCaSoChan()...
0 references
static bool XoaTatCaSoLe()...
0 references
static bool XoaTatCaSoNguyenTo()...

```

```

static bool XoaTatCaPhanTuTrungNhuau()...
0 references
static bool XoaPhanTuXuathHienNhiềuNhat()...
0 references
static bool XoaPhanTuXuathHienItNhat()...
//thay the phan tu trong mang bang phan tu moi
0 references
static bool ThayThePhanTu(int cũ, int mới)...
0 references
static int DemSoLanXuathHienCuaPhanTu(int x)...
0 references
static int DemSoDuong()...
0 references
static int DemSoAm()...
0 references
static int DemSoNguyenTo()...
0 references
static int DemSoChan()...
0 references
static int DemSoLe()...
//Kiem tra phan tu x co trong mang hay khong. Tra ve true neu co, nguoc lai false
0 references
static bool KiemTra(int x)...
0 references
static void SapXepTang()...
0 references
static void SapXepGiam()...
//Dao nguoc cac phan tu trong mang
0 references
static void DaoNguocMang()...

```