

BÀI 1.

TỔNG QUAN NGÔN NGỮ

LẬP TRÌNH C#



A. MỤC TIÊU:

- Xử lý các thao tác cơ bản trong ngôn ngữ C#.
- Sử dụng được các cấu trúc điều khiển trong C#.
- Sử dụng các vòng lặp thực hiện các bài toán cơ bản.
- Xử lý các ngoại lệ phát sinh.

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

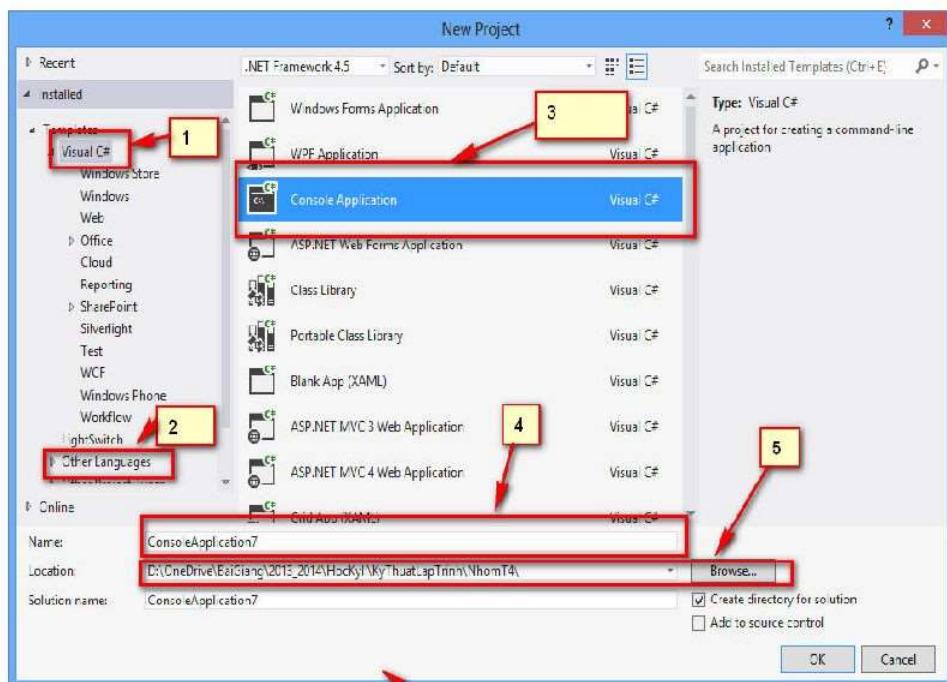
1. Cơ sở lý thuyết

1.1 Kiến thức cần nhớ

❖ Tạo một project mới

Bước 1: Khởi động Microsoft Visual Studio 2012.

Bước 2: Chọn File → New → Project



Hình 1. Tạo mới project trong Microsoft Visual Studio 2012

Trong đó:

[1][2]: Chọn loại ngôn ngữ phù hợp

[3]: Chọn loại project

[4]: Tên project

[5]: Đường dẫn lưu project

Bước 3:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Demo
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

Hình 2. Cấu trúc một chương trình C#

Trong đó:

[1]: Vùng khai báo các thư viện cần thiết.

[2]: Vùng viết các code xử lý của chương trình.

1.2 Giới thiệu bài tập mẫu

Bài 1: Viết chương trình nhập vào số nguyên, kiểm tra số đã nhập là số âm hay số dương.

Bước 1: Xử lý nhập vào số nguyên.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
```

```

static void Main(string[] args)
{
    int n; //Khai báo biến số nguyên n
    string s; //Khai báo biến s kiểu string
    //Thông báo nhập n
    Console.WriteLine("Nhập vào số nguyên n=");
    //Đọc dữ liệu nhập từ bàn phím vào chuỗi s
    s = Console.ReadLine();
    //Chuyển chuỗi s thành số
    n = int.Parse(s);
}
}

```

Hoặc có thể xử lý ngắn gọn như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n=");
            //Đọc dữ liệu nhập từ bàn phím vào chuỗi s
            //Chuyển chuỗi nhập vào thành số
            n = int.Parse(Console.ReadLine());
        }
    }
}

```

Bước 2: Kiểm tra số đã nhập là số âm hay số dương

```

..... static void Main(string[] args)
....{
..... // Khai báo biến số nguyên n
..... int n;
..... //B1
..... Console.Write("Nhập vào số nguyên n=");
..... //B2 đọc dữ liệu nhập từ bàn phím vào chuỗi s
..... //B3 Chuyển chuỗi s thành số
..... n = int.Parse(Console.ReadLine());
..... // Sử dụng cấu trúc điều kiện if
..... if (n == 0)
....{
.....     Console.WriteLine("Không âm không dương");
..... }
..... if (n > 0)
....{
.....     Console.WriteLine("Số dương");
..... }
..... if (n < 0)
....{
.....     Console.WriteLine("Số âm");
..... }
..... Console.ReadLine();
..... }
..... }

```

Trong đó:

- [1]: Sử dụng cấu trúc điều khiển if.
[2]: Dùng màn hình lại xem kết quả (tương đương getch() trong C++).
Ngoài ra chúng ta có thể sử dụng cấu trúc điều khiển đầy đủ của if như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo biến số nguyên n
            int n;
            // thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            // đọc dữ liệu nhập từ bàn phím vào chuỗi s

```

```

//Chuyển chuỗi nhập vào thành số
n = int.Parse(Console.ReadLine());
//Sử dụng cấu trúc điều kiện if
if (n == 0)
{
    Console.WriteLine("Không âm không dương");
}
else
{
    if (n > 0)
    {
        Console.WriteLine("Số dương");
    }
    else
    {
        Console.WriteLine("Số âm");
    }
}
Console.ReadLine();
}
}

```

Hướng dẫn sử dụng try...catch...

Trong một lệnh hoặc đoạn lệnh có khả năng phát sinh lỗi chúng ta nên dùng mệnh đề **try..catch..** để tránh phát sinh lỗi không mong muốn khi đang thực hiện chương trình. Ví dụ khi nhập vào số nguyên n người dùng có thể vô tình nhập một ký tự không là số khi đó lệnh:

```
n = int.Parse(Console.ReadLine());
```

sẽ xảy ra lỗi. Khi đó chúng ta có thể sử dụng mệnh đề **try..catch..** với cú pháp như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo

```

```
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            //Khai báo biến số nguyên n  
            int n;  
            //Thông báo nhập n  
            Console.WriteLine("Nhập vào số nguyên n=");  
            //Đọc dữ liệu nhập từ bàn phím vào chuỗi s  
            try  
            {  
                //Chuyển chuỗi nhập vào thành số  
                n = int.Parse(Console.ReadLine());  
            }  
            catch  
            {  
                Console.WriteLine("Bạn nhập n không phải số");  
                return;  
            }  
            // Sử dụng cấu trúc điều kiện if  
            if (n == 0)  
            {  
                Console.WriteLine("Không âm không dương");  
            }  
            else  
            {  
                if (n > 0)  
                {  
                    Console.WriteLine("Số dương");  
                }  
                else  
                {  
                    Console.WriteLine("Số âm");  
                }  
            }  
            Console.ReadLine();  
        }  
    }  
}
```

```
        }
    }
}
```

Nếu người dùng vô tình nhập một ký tự không là số thì những lệnh trong **catch{ }** sẽ được thực thi.

Bài 2: Viết chương trình tính tích: $1 * 2 * 3 * 4 * 5 * \dots * n$, trong đó n nhập từ phím.

Bộ giá trị kiểm tra:

n = 1: $S_1 = 1$;

n = 2: $S_1 = 2$;

n = 3: $S_1 = 6$

Bước 1: Xử lý nhập vào số nguyên

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n=");
            //Đọc dữ liệu nhập từ bàn phím vào chuỗi s
            //Chuyển chuỗi nhập vào thành số
            n = int.Parse(Console.ReadLine());
        }
    }
}
```

Bước 2: Xác định biểu thức 1 (bt1), biểu thức 2 (bt2), biểu thức 3 (bt3) cho vòng lặp for. Công việc cần làm là gì?

bt1 là: int i=1; // khởi tạo gán giá trị

bt2 là: i<=n; // điều kiện dừng

bt3 là: i++; // tăng bước nhảy

Công việc cần làm là tính tích giá trị từ 1 đến n. Khai báo khởi tạo biến giữ giá trị tích.

Bước 3: Code xử lý yêu cầu đề bài:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Khởi tạo giá trị cho biến tích
            int tich = 1;
            //Thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n=");
            //Nhập n từ bàn phím
            try
            {
                //Chuyển chuỗi nhập vào thành số
                n = int.Parse(Console.ReadLine());
            }
            catch
            {
                Console.WriteLine("Bạn nhập n không phải số");
                return;
            }
            //Xử lý tính tích
            for (int i = 1; i <= n; i++)
            {
                tich = tich * i;
            }
            //In giá trị ra màn hình theo 1 trong 2 cách
            Console.WriteLine("Tích là: {0}",tich);
        }
    }
}
```

```

        //Console.WriteLine("Tich la: " + tich);
        Console.ReadLine();
    }
}
}

```

Chúng ta có thể chuyển ví dụ trên thành dạng phương thức gọi khi cần. Lưu ý có thể đặt phương thức phía trên hoặc phía dưới void main() mà không cần phải khai báo prototype (nguyên mẫu hàm).

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Gọi phương thức BaiMau()
            BaiMau();
            Console.ReadLine();
        }
        //Phương thức BaiMau() được định nghĩa như sau
        static void BaiMau()
        {
            //Khai báo biến số nguyên n
            int n;
            //Khởi tạo giá trị cho biến tích
            int tich = 1;
            // thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            //Nhập từ bàn phím
            try
            {
                // Chuyển chuỗi nhập vào thành số

```

```

        n = int.Parse(Console.ReadLine());
    }
    catch
    {
        Console.WriteLine("Bạn nhập n không phải số");
        return;
    }
    // Xử lý tính tích
    for (int i = 1; i <= n; i++)
    {
        tich = tich * i;
    }
    // In giá trị ra màn hình
    Console.WriteLine("Tích la: {0}", tich);
    //Console.WriteLine("Tích la: " + tich);
}
}

```

Bài 3: Viết chương trình tìm ước chung lớn nhất (UCLN) của hai số nguyên a và b.

Bước 1: Xử lý nhập vào 2 số nguyên a và b

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo
            int a, b;
            //Nhập a, b
            Console.WriteLine("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());

```

```

        Console.WriteLine("Mời nhập b: ");
        b = int.Parse(Console.ReadLine());
        Console.ReadLine();
    }
}
}

```

Bước 2: Xác định biểu thức 1, biểu thức 2, biểu thức 3 cho bài toán. Công việc cần làm là gì?

Bước 3: Công việc cần làm là tìm UCLN của 2 số a và b, thực hiện như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo
            int a, b;
            // Nhập a, b
            Console.WriteLine("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());
            Console.WriteLine("Mời nhập b: ");
            b = int.Parse(Console.ReadLine());
            while (a != b) //bt2
            {
                if (a > b)
                {
                    a = a - b; //bt2
                }
                else
                {
                    b = b - a; //bt2
                }
            }
        }
    }
}

```

```

        }
    }
    // In kết quả
    Console.WriteLine("UCLN là: ", a);
    Console.ReadLine();
}
}
}

```

Viết dạng phương thức bài trên chúng ta làm như sau:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo
            int a, b;
            // Nhập giá trị cho biến a và b
            Console.Write("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());
            Console.Write("Mời nhập b: ");
            b = int.Parse(Console.ReadLine());
            //In kết quả
            Console.WriteLine("UCLN là: ", UCLN(a, b));
            Console.ReadLine();
        }
        static int UCLN(int a, int b)
        {
            while (a != b) //bt2
            {

```

```

    if (a > b)
    {
        a = a - b; //bt2
    }
    else
    {
        b = b - a; //bt2
    }
}
return a;
}
}

```

2 . Bài tập tại lớp

Bài 1. Viết chương trình nhập vào đơn giá một mặt hàng, và số lượng bán của mặt hàng. Tính tiền khách phải trả, với thông tin như sau:

- Thành tiền: đơn giá * số lượng
- Giảm giá: Nếu thành tiền > 100, thì giảm 3% thành tiền, ngược lại không giảm
- Tổng tiền phải trả: thành tiền – giảm giá.

Bài 2. Viết chương trình tính tiền điện phải trả trong tháng:

- Từ 01 – 100KW: 5\$
- Từ 101 – 150KW: 7\$
- Từ 151 – 200KW: 10\$
- Từ 201 – 300KW: 15\$
- Từ 301KW trở lên: 20\$

Ví dụ: Nếu điện tiêu thụ 50KW thì tiền điện phải trả là: $50 \times 5 = 250$.

Nếu điện tiêu thụ 101KW thì tiền điện phải trả là: $100 \times 5 + 1 \times 7 = 507$.

Bài 3. Viết chương trình in bảng cửu chương từ 1 đến 9 theo hàng dọc.

Bài 4. Viết chương trình in bảng cửu chương từ 1 đến 9 theo hàng ngang.

Bài 5. Viết chương trình vẽ hình chữ nhật có kích thước $d \times r$, trong đó d là chiều dài, và r là chiều rộng được nhập từ phím.

```

*   *
*   *
*   *

```

Bài 6. Viết chương trình vẽ hình chữ nhật có kích thước $d \times r$, trong đó d là chiều dài, và r là chiều rộng được nhập từ phím.

```
*      *      *      *      *
*                      *
*      *      *      *      *
```

Bài 7. Viết chương trình tính tổng: $1^2 + 2^2 + \dots + n^2$ với $n > 0$ nhập từ bàn phím.

Bộ giá trị kiểm tra:

$$n = 1: S_1 = 1; \quad n = 2: S_1 = 5; \quad n = 3: S_1 = 14$$

Bài 8. Kiểm tra n có phải là số nguyên tố hay không?

Bài 9. Liệt kê các số nguyên tố $< n$

Bài 10. Nhập dãy số nguyên (nhập số 0 thì dừng)

- a. Tìm tổng dãy số vừa nhập
- b. Tìm max của dãy số đó
- c. Tìm min của dãy số đó

3 . Bài tập nâng cao

Bài 1. Viết chương trình hỗ trợ cách giải phương trình bậc 2 ($ax^2 + bx + c = 0$).

Bài 2. In tam giác cân rộng.

4 . Bài tập về nhà

Bài 1. Tính giá trị hàm

$$\begin{aligned} \text{a. } f(x) &= \begin{cases} 3x + \sqrt{x} & , x > 0 \\ e^x + 4 & , x \leq 0 \end{cases} \\ \text{b. } f(x) &= \begin{cases} \sqrt{x^2 + 1} & , x \geq 1 \\ 3x + 5 & , -1 < x < 1 \\ x^2 + 2x - 1 & , x \leq -1 \end{cases} \end{aligned}$$

Gợi ý dùng thư viện **Math.Sqrt**, **Math.Pow**, **Math.E**

Bài 2. Viết chương trình xuất số có 2 chữ số sao cho các chữ số khác nhau đỏi một.

Ví dụ: danh sách các số thỏa yêu cầu:

Danh sach cac so co 2 chu so doi mot khac nhau la: 10, 12, 13, ..., 20, 21, 23, ..., 98

Bài 3. Viết chương trình in ra tất cả các số lẻ nhỏ hơn n , trong đó n nhập từ bàn phím.

Bài 4. Viết chương trình đếm số lượng số chẵn trong $[n, m]$, trong đó n, m nhập từ bàn phím.

Trường ĐH CNTP TP.HCM Khoa: CNTT Bộ môn: KTPM Môn: Công nghệ .NET	BÀI 2. MẢNG, MẢNG CHUỖI	
--	--	--

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`”. Ưu điểm 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.WriteLine("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.WriteLine("\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.WriteLine("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.WriteLine("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.Write("Mời bạn nhập số dòng: ");
    n = int.Parse(Console.ReadLine());
    Console.Write("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.WriteLine("Chuoi doi xung");
    }
    else
    {
        Console.WriteLine("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 $m * n$. 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 $m * n$. 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.