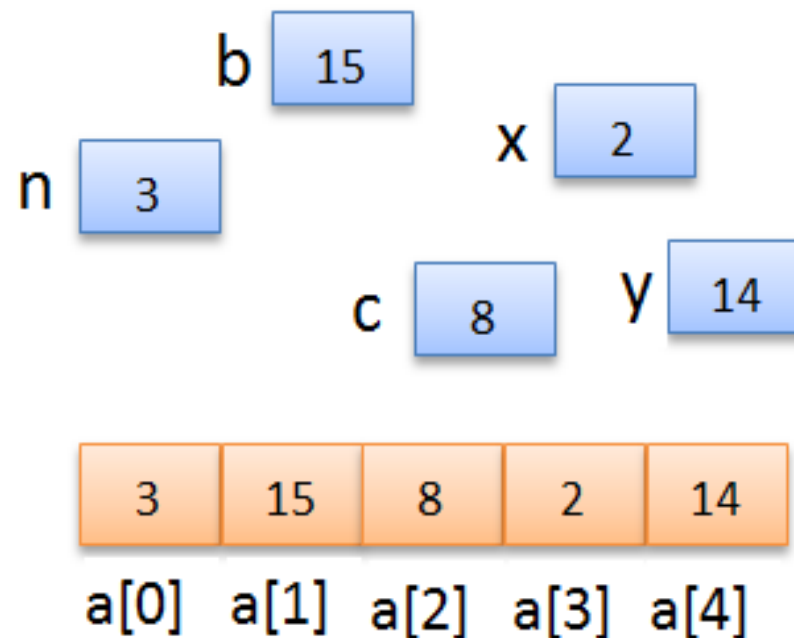


CHƯƠNG 2

KIỂU MẢNG VÀ XÂU KÝ TỰ



Giảng viên: **Nguyễn Quỳnh Diệp – Khoa CNTT – ĐH Thủy Lợi**

Email: diepngq@tlu.edu.vn

NỘI DUNG

1. Kiểu mảng
2. Kiểu cấu trúc

KIỂU MẪNG

NỘI DUNG

1. Mảng 1 chiều
2. Mảng 2 chiều
3. Một số thao tác trên mảng 1 chiều (Tìm kiếm, sắp xếp, chèn, xóa,...)
4. Truyền tham số là kiểu mảng

KHÁI NIỆM

- **Mảng** là tập hợp hữu hạn các phần tử có cùng kiểu dữ liệu
- Các phần tử lưu trữ ở các vị trí kế tiếp nhau trong bộ nhớ
- Kích thước của mảng bằng số phần tử
- Mảng một chiều, mảng hai chiều

0	1	2	3	4	5
6	23	12	9	14	52
0	0	0	0	0	0
x	x	x	x	x	x
0	0	0	0	0	0
1	1	1	1	1	1
2	3	4	5	6	7

MẢNG 1 CHIỀU

KHAI BÁO MẢNG MỘT CHIỀU

■ Cú pháp: `kieudulieu tenmang [sophantu tối đa];`

- Phần tử của mảng được truy cập thông qua tên mảng
- Các phần tử được đánh số từ **0** → **(số phần tử - 1)**
- Số phần tử là hằng số

■ Ví dụ:

```
int x[10];  
  
char hoten[50];  
  
float a[10], b[20];
```

KHỞI TẠO GIÁ TRỊ CHO MẢNG

▪ Cú pháp:

kieudulieu tenmang [sophantu] = {giatri1, giatri2...., giatriN};



Chú ý:

- Các giá trị khởi tạo phải là **hằng**
- Nếu số lượng giá trị khởi tạo ít hơn số phần tử thì các phần tử còn lại sẽ có giá trị 0
- Có thể không cần khai báo kích thước của mảng khi khởi tạo

KHỞI TẠO GIÁ TRỊ CHO MẢNG

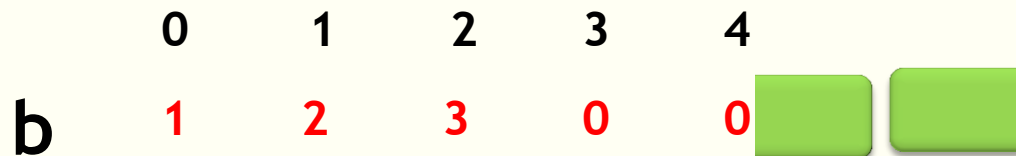
- Ví dụ 1: Khởi tạo cho mọi phần tử của mảng

```
char a[5] = {'A', 'B', 'C', 'D', 'E'};
```



- Ví dụ 2: Khởi tạo cho một số phần tử của mảng

```
int b[5] = {1, 2, 3};
```



NHẬP GIÁ TRỊ CHO CÁC PHẦN TỬ CỦA MẢNG

➤ Cách 1: Nhập trực tiếp thông qua chỉ số mảng

```
b[0] = 1;
```

```
b[1] = 2;
```

➤ Cách 2: Nhập thông qua vòng lặp **for**

```
for(int i=0; i< n; i++)  
{  
    cout<<"a["<<i<<"]=";  
    cin>>a[i];  
}
```



Chỉ số phải nhỏ hơn số phần tử của mảng

XUẤT GIÁ TRỊ CÁC PHẦN TỬ CỦA MẢNG

➤ Cách 1: Xuất thông qua chỉ số mảng

```
int a = b[0];  
cout<<b[1];
```

➤ Cách 2: Xuất thông qua vòng lặp **for**

```
for(int i=0; i< n; i++)  
{  
    cout<<"a["<<i<<"]= ";  
    cout<<a[i]<<endl;  
}
```

VÍ DỤ

- Đọc vào dãy n số nguyên ($n \leq 100$)
- Hiển thị dãy số vừa nhập ra màn hình

VÍ DỤ

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n, a[100];
6      cout<<"nhap so phan tu cua day n=";cin>>n;
7      for(int i=0;i<n;i++)
8      {
9          cout<<"a["<<i<<"]=";
10         cin>>a[i];
11     }
12     cout<<"Day so vua nhap"<<endl;
13     for(int i=0;i<n;i++)
14         cout<<a[i]<<" ";
15     return 0;
16 }
```



BÀI TẬP

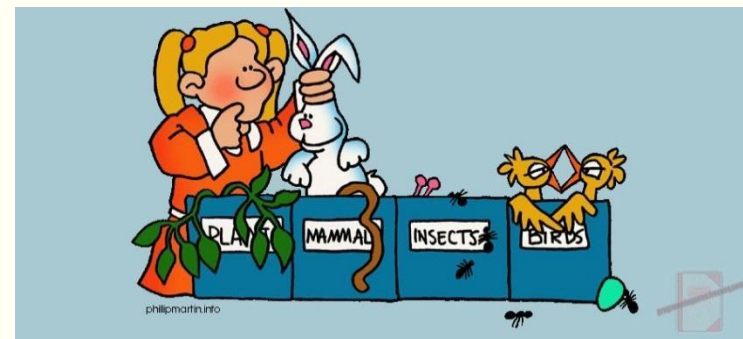
- Bài 1: Nhập vào 1 dãy gồm n số nguyên $A_1, A_2 \dots A_n$ từ bàn phím ($0 < n \leq 20$).
 - a. In ra các số trong dãy là số chẵn
 - b. Đếm các số trong dãy lớn hơn 10
 - c. Tìm các số bằng số trước nó cộng 2
 - d. Tìm các bộ ba số trong dãy thỏa mãn điều kiện

$$A_i = A_{i-1} + A_{i+1}$$



MỘT SỐ BÀI TOÁN TRÊN MẢNG

- Tìm giá trị lớn nhất, nhỏ nhất trong dãy số.
- Sắp xếp dãy số tăng dần (giảm dần)
- Chèn thêm một số vào dãy số ban đầu
- Xóa một phần tử của dãy số



TÌM GIÁ TRỊ LỚN NHẤT

- Bài toán:

- Nhập vào dãy n số nguyên ($n \leq 20$)
- Đưa ra màn hình giá trị lớn nhất trong dãy vừa nhập

TÌM GIÁ TRỊ LỚN NHẤT

```
int main ()
{
    int a[20], n;
    cout<<"Nhap so phan tu cua day (<100) = ";
    cin>>n;
    for(int i=0; i<n; i++)
    {
        cout<<"a["<<i<<"= ";
        cin>>a[i];
    }
    int max = a[0];
    for(int i=1; i<n; i++)
        if(a[i] > max) max = a[i];
    cout<<"Gia tri lon nhat cua day la: "<<max;
    return 0;
}
```

SẮP XẾP MẢNG

■ Bài toán:

Nhập vào một mảng gồm n phần tử.

Sắp xếp các phần tử của mảng theo thứ tự tăng dần.

Hiển thị mảng đã sắp xếp ra màn hình.

12	9	24	35	11	18
----	---	----	----	----	----



9	11	12	18	24	35
---	----	----	----	----	----

SẮP XẾP MẢNG

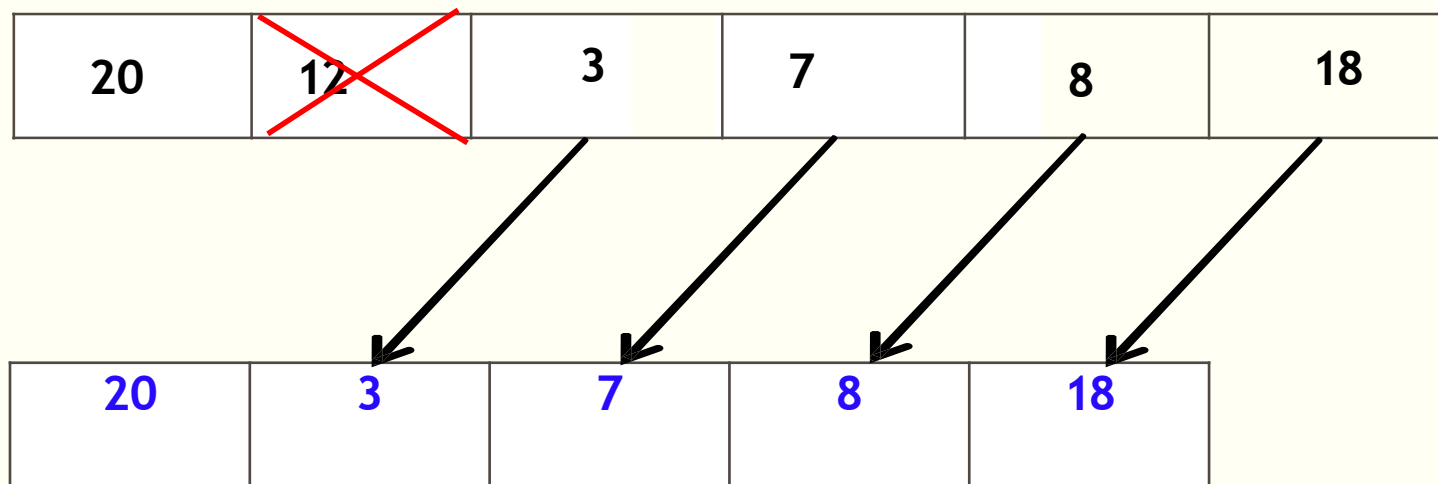
```
int main ()
{   int a[100],n;
    cout<<"Nhap so phan tu cua day (<100) = "; cin>>n;
    for(int i=0; i<n; i++)
    { cout<<"a["<<i<<"= ";
      cin>>a[i];
    }
    for(int i=0;i<n-1;i++)
        for(int j=i+1; j<n; j++)
            if(a[j] < a[i])
            { int tg=a[i];
              a[i]=a[j];
              a[j]=tg;
            }
    cout<<"Sau khi sap xep:"<<endl;
    for(i=0; i<n; i++)
        cout<<a[i]<<" ";
    return 0;
}
```

a0	a1	a2	a3	a4
20	3	7	5	18
3	20	7	5	18
	7	20	5	18
	5	20	7	18
		7	20	18
			18	20

XÓA 1 PHẦN TỬ KHỎI MẢNG

▪ Bài toán:

- Lập trình nhập vào một mảng gồm n phần tử.
- Nhập vào một giá trị cần xóa khỏi mảng.
- Hiển thị mảng sau khi đã xóa.



XÓA 1 PHẦN TỬ KHỎI MẢNG

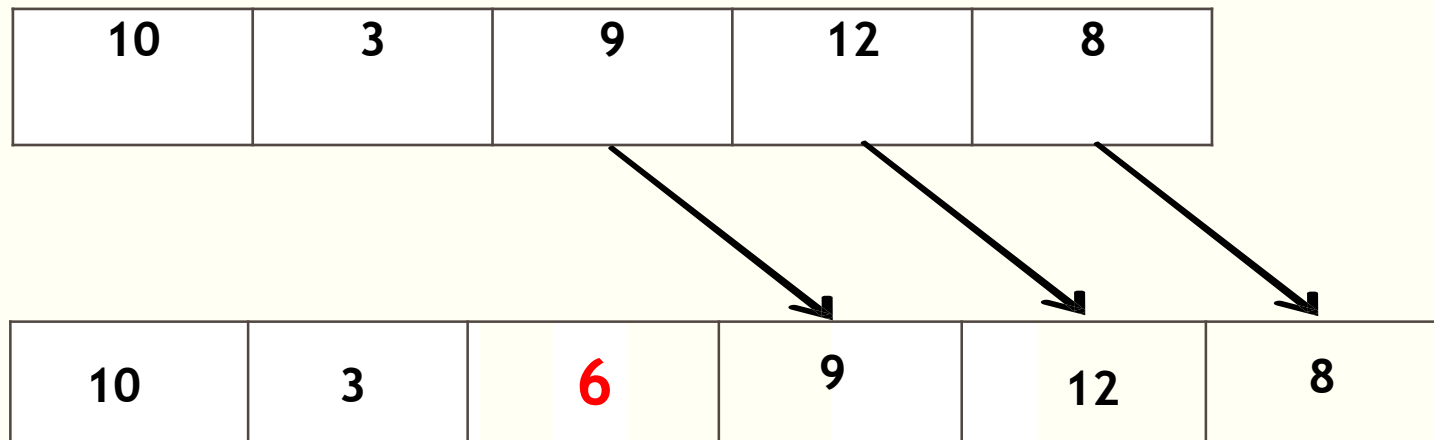
```
int main ()
{
    int a[100], n, x;
    cout<<"Nhap so phan tu cua day (<100) =";
    cin>>n;
    for(int i=0; i<n; i++)
    {
        cout<<"a["<<i<<"="; cin>>a[i];
    }
    cout<<"Gia tri can xoa:";
    cin>>x;
    int i=0; m=n;
    while(i<m)
        if(a[i]==x)
        {
            for(int j=i; j<m-1; j++)
                a[j] = a[j+1];
            m--;
        }
        else
            i++;
}
```

```
if(m==n)
    cout<<"Khong co "<<x;
else
{
    cout<<"Day moi la: "<<endl;
    for(i=0; i<m; i++)
        cout<<a[i]<<" ";
}
return 0;
}
```

CHÈN THÊM 1 PHẦN TỬ VÀO MẢNG

▪ Bài toán:

- Lập trình nhập vào một mảng gồm n phần tử.
- Nhập vào một giá trị cần thêm vào mảng.
- Nhập vị trí (chỉ số) cần thêm vào mảng
- Hiển thị mảng sau khi đã thêm



CHÈN THÊM 1 PHẦN TỬ VÀO MẢNG

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  int main ()
5  {
6      int a[100], n, k, x;
7      cout<<"Nhap so phan tu cua day n = "; cin>>n;
8      for(int i=0; i<n; i++)
9      {
10         cout<<"a["<<i<<" ] = "; cin>>a[i];
11     }
12     cout<<"Vi tri can chen: "; cin>>k;
13     cout<<"\nNhap gia tri can chen x = "; cin>>x;
14     //Chen phan tu x vao vi tri k
15     for(int i=n;i>=k;i--)
16     {
17         a[i]=a[i-1];
18     }
19     a[k]=x;
20     n++;
21     //In mang
22     cout<<"Mang sau khi chen la: ";
23     for(int i=0; i<n; i++)
24         cout<<a[i]<<" ";
25     return 0;
26 }
```

TRUYỀN THAM SỐ LÀ MẢNG

- Một phần tử của mảng có thể làm đối số cho 1 hàm
VD: binhphuong(a[4])
- Một mảng cũng có thể làm tham số cho 1 hàm
VD: void sapxep(int a[], int n);
- Khi gọi hàm có tham số là 1 mảng chỉ cần truyền tên mảng (không cần truyền kích cỡ mảng), đối số truyền vào sẽ là địa chỉ đầu mảng
VD: sapxep(a, 5);

TRUYỀN THAM SỐ LÀ MẢNG

Định nghĩa hàm

Truyền mảng vào hàm

```
#include <iostream> using
namespace std; void
nhap(int a[],int n)
{
    for(int i=0; i<n; i++)
    {
        cout<<"a["<<i<<"="; cin>>a[i];
    }
}
void hienThi(int a[],int n)
{
    for(int i=0; i<n; i++)
        cout<<"a["<<i<<"="<<a[i]<<endl;
}
int main ()
{
    int a[100], n;
    cout<<"Nhap so phan tu cua day (<100) ="; cin>>n;
    nhap(a, n) ;
    cout<<"Day so vua nhap:"<<endl;
    hienThi(a, n);
    return 0;
}
```



BÀI TẬP

Bài 2: Nhập n là số nguyên dương ≤ 20 .

- Nhập mảng A gồm n phần tử.
- In ra mảng A đã nhập
- Tính tổng các phần tử ở chỉ số lẻ
- Sắp xếp mảng A theo thứ tự giảm dần. Sau đó gọi hàm In mảng ở câu b để in mảng sau khi sắp xếp.
- Nhập 1 số x từ bàn phím, tìm trong mảng xem có phần tử nào bằng x không? Xóa phần tử x khỏi mảng. In ra mảng sau xóa.
- Nhập số $k > 0$. Thêm vào vị trí k trong mảng 1 phần tử có giá trị x được nhập từ bàn phím. In mảng sau khi thêm.



Chú ý: Mỗi câu viết bằng 1 hàm

BÀI TẬP

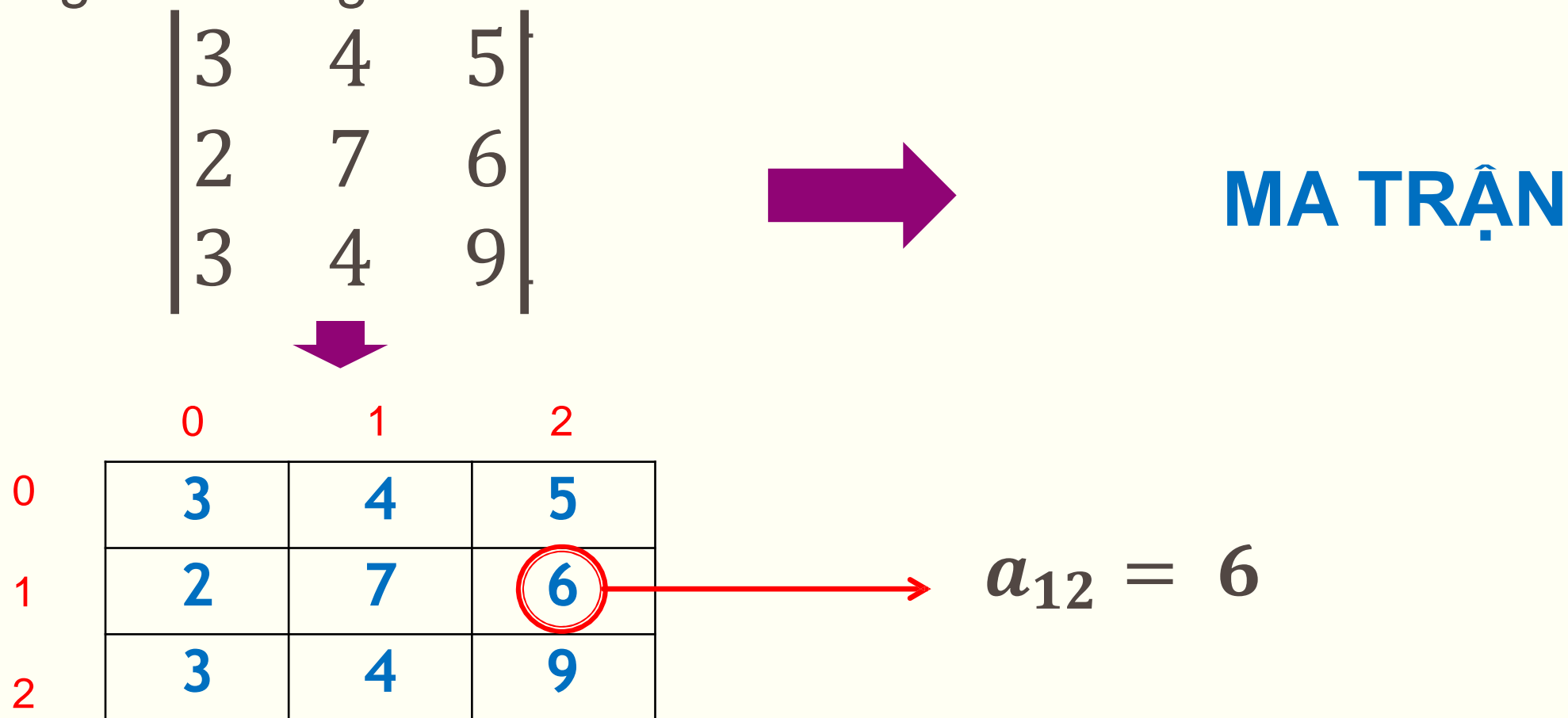
Bài 4 : Viết chương trình nhập dãy A có n số thực. Nhập vào một số thực x , Tách dãy A thành 2 dãy con, dãy thứ nhất gồm toàn những phần tử nhỏ hơn x , dãy thứ hai gồm những phần tử còn lại. In hai dãy con ra màn hình.

Bài 5 : Đa thức bậc n : $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ có thể được lưu trữ trong một vector (mảng) các hệ số $(a_0, a_1, a_2, \dots, a_n)$. Hãy viết chương trình nhập các hệ số một đa thức, và tính giá trị của đa thức bậc n trên theo giá trị x (cũng được nhập từ bàn phím).

MẢNG 2 CHIỀU

MẢNG 2 CHIỀU

- Mảng dữ liệu có thể có 2 chiều. Mảng 2 chiều được coi là mảng của mảng.



KHAI BÁO MẢNG HAI CHIỀU

▪ Cú pháp: `kieudulieu tenmang [sohang][socot];`

▪ Ví dụ:

```
int x[3][5];  
double a[10][20];
```

▪ Khởi tạo giá trị:

```
int x[3][4] = {1, 3, 5, 7, 2, 4, 6, 8, 10, 20, 30, 40}  
int x[3][2] = { {1, 2} , {3, 4} , {5, 6} }  
int a[][] = { {2, 4, 5}, {3, 5, 9} , {7, 2, 8} }
```

NHẬP XUẤT MẢNG 2 CHIỀU

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  int main ()
5  { int hang, cot;
6    int a[10][20];
7    //nhap mang
8    cout << "Nhap so hang cua mang: "; cin >> hang;
9    cout << "Nhap so cot cua mang: "; cin >> cot;
10   cout << "Nhap cac phan tu cua mang: \n";
11   for (int i=0; i<hang; i++)
12       for (int j=0; j<cot; j++)
13       {
14           cout << "a[" << i << "][" << j << "] = " ;
15           cin >> a[i][j] ;
16       }
17   // xuat mang;
18   cout << "Mang da nhap la: \n";
19   for (int i=0; i<hang; i++)
20   {
21       for (int j=0; j<cot; j++)
22           cout << a[i][j] << " ";
23       cout << endl;
24   }
25 }
```

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  int main ()
5  { int hang, cot;
6    int a[10][20];
7    //nhap mang
8    cout << "Nhap so hang cua mang: "; cin >> hang;
9    cout << "Nhap so cot cua mang: "; cin >> cot;
10   cout << "Nhap cac phan tu cua mang: \n";
11   for (int i=0; i<hang; i++)
12       for (int j=0; j<cot; j++)
13       {
14           cout << "a[" << i << "][" << j << "] = " ;
15           cin >> a[i][j] ;
16       }
17   // xuat mang;
18   cout << "Mang da nhap la: \n";
19   for (int i=0; i<hang; i++)
20   {
21       for (int j=0; j<cot; j++)
22           cout << a[i][j] << " ";
23       cout << endl;
24   }
25 }

```

	0	1	2
0	3	4	5
1	2	7	6
2	3	4	9

CỘNG 2 MẢNG HAI CHIỀU

Chỉ cộng được 2 mảng cùng kích thước:

$$\begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 1 & 3 \\ 2 & 3 & 1 \\ 1 & 3 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 5 \\ 4 & 4 & 4 \\ 4 & 5 & 3 \end{bmatrix}$$

CỘNG 2 MẢNG HAI CHIỀU

```
#include <iostream>
using namespace std;

int main ()
{
    int a[30][30], b[30][30], c[30][30];
    int i, j, cot, hang;
    cout<<"Nhap so hang, cot cua ma tran: ";
    cin>>hang>>cot;
    cout<<"Nhap ma tran A"<<endl;
    for(i=0; i< hang; i++)//nhap gia tri cho mang A
        for(j=0; j<cot; j++)
        {
            cout<<"a["<<i<<"]["<<j<<"]="; cin>>a[i][j];
        }
    cout<<"Nhap ma tran B"<<endl;
    for(i=0; i< hang; i++)//nhap gia tri cho mang B
        for(j=0; j<cot; j++)
        {
            cout<<"b["<<i<<"]["<<j<<"]="; cin>>b[i][j];
        }
```

```
    cout<<"\nMa tran tong la:"<<endl;
    for(i=0; i< hang; i++)//ma tran tong
    {
        for(j=0; j<cot; j++)
        {
            c[i][j]= a[i][j] + b[i][j];
            cout<<c[i][j]<<" ";
        }
        cout<<endl;
    }
}
```



BÀI TẬP

Bài 3: Viết chương trình nhập vào 1 mảng A kích thước 4x4 gồm các số nguyên.

- Đưa ra các giá trị >5 trên dòng $i=3$ (dòng cuối)
- Đưa ra số lượng các số âm và dương trên cột $j=2$ của mảng
- Tính tổng các phần tử trên đường chéo chính của mảng
- Tính tổng và trung bình cộng của các phần tử trên hàng đầu tiên ($i=0$)

Chú ý: Sử dụng hàm để viết chương trình trên





BÀI TẬP

Bài 4:

a. Viết chương trình nhập mảng A có kích thước $m \times n$

($2 \leq m \leq 5$, $2 \leq n \leq 10$). In ra mảng A theo dạng hàng, cột

b. Xây dựng mảng S có kích thước $m \times n$ tương ứng, trong đó:

$$s_{i,j} = \begin{cases} 1 & \text{nếu } a_{i,j} > 0 \\ 0 & \text{nếu } a_{i,j} = 0 \\ -1 & \text{nếu } a_{i,j} < 0 \end{cases}$$

In ra màn hình mảng S theo dạng ma trận (hàng, cột)





BÀI TẬP

Bài 5: Viết chương trình nhập vào 1 mảng A kích thước $n \times m$ ($0 < n, m \leq 15$) gồm các số thực.

- Nhập mảng và xuất mảng ra màn hình
- Tìm giá trị nhỏ nhất trên dòng $i = 2$
- Nhập giá trị x từ bàn phím. Tìm xem trên cột 1 có phần tử nào bằng x không? Nếu có thì đưa ra vị trí (chỉ số) của phần tử đó.
- Sắp xếp cột cuối cùng theo thứ tự tăng dần
- Đổi chỗ các phần tử trên hàng đầu và hàng cuối cho nhau.
- In ra các số nguyên tố trong mảng



KIỂU XÂU KÝ TỰ

**KHÁI NIỆM, KHAI BÁO,
NHẬP XUẤT XÂU KÝ TỰ**

KHÁI NIỆM

- **Xâu** là một mảng (dãy) các ký tự
- Khi lưu trữ xâu ký tự kết thúc bằng ký tự **'\0'**
- **Ví dụ:** “Ngôn ngữ C++”

0	1	2	3	4	5	6	7	8	9	10	11	12	13
N	g	o	n		n	g	u		C	+	+	\0	



Kí tự đặt trong dấu nháy đơn ‘ ‘

Xâu kí tự đặt trong dấu nháy kép “ “

KHAI BÁO XÂU KÍ TỰ

▪ Cách 1: `char tenxau [chieudai];`

`char tenxau [chieudai] = <Chuỗi kí tự>;`

`char tenxau [chieudai] = { 'kí tự 1', 'kí tự 2', '\0' };`

▪ Ví dụ:

`char xau[15] = "Xin chao!";`

`char xau[15] = {'H', 'e', 'l', 'l', 'o', '\0'};`

`char xau[] = "Xin chao!";`

KHAI BÁO XÂU KÍ TỰ

▪ Cách 2: sử dụng lớp **string**

```
string tenxau;
```

```
string tenxau = <Chuỗi kí tự>;
```

▪ Ví dụ:

```
string str;
```

```
str="Xin chao!";
```

```
string xau = "Xin chao!";
```

TRUY NHẬP VÀO PHẦN TỬ CỦA XÂU

▪ Cú pháp:

tenxau [chỉ số của kí tự]

▪ Ví dụ:

```
string str = "Ha Noi";
```

```
str[0] → lưu 'H'
```

```
str[1] → lưu 'a'
```

```
str[2] → lưu ' '
```

MỘT SỐ HÀM VỚI KÍ TỰ

Hàm	Mô tả
tolower(ch)	Chuyển thành kí tự thường
toupper(ch)	Chuyển thành kí tự hoa
islower(ch)	Kiểm tra chữ thường
isupper(ch)	Kiểm tra chữ hoa
isdigit(ch)	Kiểm tra chữ số
isalpha(ch)	Kiểm tra xem kí tự có là chữ cái không
isspace(ch)	Kiểm tra kí tự dấu cách
isctrl(ch)	Kiểm tra kí tự điều khiển

MỘT SỐ VÍ DỤ

Ví dụ 1: Nhập 1 xâu và đếm số ký tự viết hoa

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main()
5  {
6      string str;
7      cout<<"Nhap vao mot xau ki tu:";
8      getline(cin, str);
9      int i=0, dem=0;
10     while(str[i]!='\0')
11     {
12         if(isupper(str[i]))//Kiem tra neu la ki tu hoa
13             dem++;
14         i++;
15     }
16     cout<<"So ki tu hoa la:"<<dem;
17     return 0;
18 }
```

MỘT SỐ VÍ DỤ

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main()
5  {
6      string str;
7      cout<<"Nhap vao mot xau ki tu:";
8      getline(cin, str);
9      int i, dem=0;
10     for(i=0;i<=str.size(); i++)
11     {
12         if(isupper(str[i]))//Kiem tra neu la ki tu hoa
13             dem++;
14     }
15     cout<<"So ki tu hoa la:"<<dem;
16     return 0;
17 }
```

MỘT SỐ VÍ DỤ

- Ví dụ 2: Nhập vào một chuỗi ký tự. Chuyển chuỗi đó thành chữ thường.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main ()
6  {
7      string s;
8      cout<<"Nhập vào một chuỗi: "<<endl;
9      getline(cin, s);
10
11     for(int i=0; i<s.size();i++)
12         s[i] = tolower(s[i]);
13
14     cout<<"\nChuyển thành chuỗi thường:\n"<< s;
15     return 0;
16 }
```

MỘT SỐ VÍ DỤ

Ví dụ 3: Nhập vào một chuỗi kí tự và một kí tự. Đếm số lần xuất hiện của kí tự đó trong chuỗi.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main ()
5  {
6      string s; char c;
7      cout<<"Nhập vào một chuỗi: "<<endl;
8      getline(cin, s);
9      cout<<"Nhập vào một kí tự ";
10     cin>>c;
11     int dem =0;
12     for(int i=0; i<s.size();i++)
13         if(s[i]==c)
14             dem++;
15
16     cout<<"\nSố lần xuất hiện kí tự "<<c<<" là:"<<dem;
17     return 0;
18 }
```


MỘT SỐ THAO TÁC TRÊN XÂU KÝ TỰ

- So sánh xâu
- Ghép xâu
- Tìm kiếm
- Xóa xâu con
- Thêm xâu con

CÁC PHÉP TOÁN VÀ PHƯƠNG THỨC CƠ BẢN

Phép toán/Phương thức	Mô tả
+	Ghép 2 chuỗi xâu hoặc ghép một kí tự vào xâu
==, !=, >, >=, <, <=	So sánh theo thứ tự từ điển
.length(), .size()	Trả về độ dài của xâu ban đầu
.append(str)	Thêm xâu str vào cuối xâu ban đầu
.replace(pos, len, str)	Thay thế len ký tự trong ban đầu, bắt đầu từ vị trí pos bằng 1 xâu con str
.substr(pos, len)	Trích xâu con dài len ký tự bắt đầu từ vị trí pos của xâu ban đầu
.compare(str)	So sánh xâu str với xâu ban đầu

SO SÁNH XÂU

■ Toán tử so sánh

```
string str = "FILENAME";  
str == "FILENAME"    → Cho giá trị True  
str < "FILENAME A"  → Cho giá trị True
```

■ Dùng phương thức **compare**

```
string str1 = "FILENAME";  
string str2 = "FILENAME";  
str1.compare(str2) → Cho giá trị = 0 nếu str1=str2  
                  → Cho giá trị > 0 nếu str1 > str2  
                  → Cho giá trị < 0 nếu str1 < str2
```

GHÉP XÂU

▪ Dùng toán tử '+'

```
string str1 = "Hello. ";  
string str2;  
str2 = "How are you?";  
str1 = str1 + str2;  
cout<<str1;
```

Hello. How are you?

▪ Dùng phương thức append

```
string str1 = "Hello. ";  
str1.append("How are you?");  
cout<<str1;
```

Hello. How are you?

CHÈN XÂU

Cú pháp	Mô tả
.insert(pos, str2)	Chèn xây str2 vào vị trí pos trong xâu ban đầu
insert(pos, str2, subpos, sublen)	Chèn 1 xâu con của xâu str2 vào vị trí pos của xâu ban đầu. Xâu con của str2 bắt đầu lấy từ vị trí subpos và chiều dài xâu con là sublen
insert(pos, n, c)	Chèn n lần ký tự c vào vị trí pos của xâu ban đầu

XÓA XÂU

Cú pháp	Mô tả
. erase (pos, len)	Xóa xâu ban đầu len ký tự, bắt đầu xóa từ vị trí pos
. clear()	Xóa xâu ban đầu

TÌM KIẾM XÂU

Cú pháp	Mô tả
<code>. find(str)</code>	Tìm xâu str trong xâu ban đầu. Trả ra vị trí đầu tiên tìm thấy str .
<code>. find(str, pos)</code>	Tìm xâu str trong xâu ban đầu, bắt đầu tìm từ vị trí pos . Trả ra vị trí đầu tiên tìm thấy str , tính từ pos

VÍ DỤ

Nhập vào một xâu kí tự và một xâu con. Đếm số lần xuất hiện của xâu con đó trong xâu.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  int main ()
5  {
6      string s1, s2;
7      cout<<"Nhập vào một xâu gốc:"; getline(cin,s1);
8      cout<<"Nhập vào một xâu con:"; getline(cin, s2);
9      int i=0, dem=0, pos;
10     while(i<s1.size())
11     {
12         pos = s1.find(s2,i);
13         if(pos<0)
14             break;
15         else
16         {
17             dem++;
18             i=pos + s2.size();
19         }
20     }
21     cout<<"Số lần xuất hiện của xâu '"<<s2<<"' là:"<<dem;
22     return 0;
23 }
```




BÀI TẬP

- **Bài 1:** Viết chương trình nhập vào một chuỗi x và ký tự c. Kiểm tra xem ký tự c có xuất hiện trong chuỗi x hay không? Nếu có thì xuất hiện bao nhiêu lần, đưa ra vị trí của c.
- **Bài 2:** Viết chương trình nhập vào một chuỗi và đưa ra màn hình thông báo chuỗi đó có **đối xứng** hay không.





BÀI TẬP

- **Bài 3:** Cho chuỗi x và chuỗi y. Viết chương trình nhập vào 2 giá trị p và q từ bàn phím. Chèn chuỗi y vào chuỗi x tại các vị trí p và q. Đưa kết quả chuỗi x sau khi chèn ra màn hình.
- **Bài 4:** Một chuỗi được gọi là chuỗi chuẩn nếu đầu và cuối chuỗi không chứa dấu cách, đồng thời trong chuỗi không chứa hai dấu cách liên nhau.

Viết chương trình nhập vào một chuỗi và đưa ra màn hình chuỗi **chuẩn** của chuỗi đó.

