



FUTURE TECHNOLOGY
CÔNG NGHỆ TƯƠNG LAI

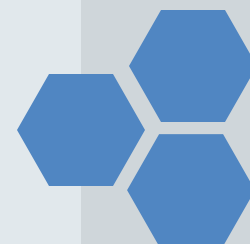
TRƯỜNG ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP
Khoa Công Nghệ Thông Tin

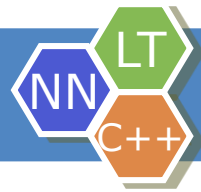


Phần 2. NN lập trình C++



CHƯƠNG 6 MẢNG VÀ XÂU KÝ TỰ





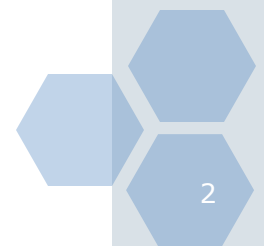
Chương 6 – Mảng và chuỗi ký tự

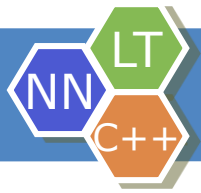


Mảng



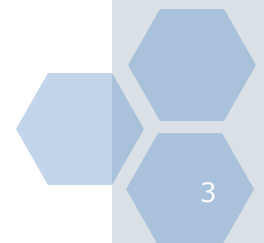
Xâu ký tự

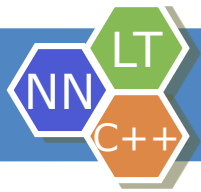




6.1 Mảng

- Mảng (array)
 - Tập hợp các phần tử có cùng kiểu dữ liệu và có số lượng xác định.
 - Thực thể tĩnh (giữ nguyên kích thước trong suốt chương trình)
- Một vài loại mảng
 - mảng dựa vào con trỏ (Pointer-based arrays)
 - mảng là đối tượng (Arrays as objects) (C++)

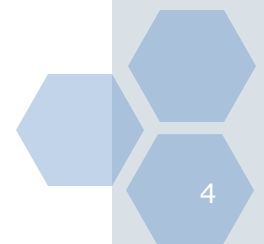




6.1 Mạng

6.1.1 Mạng
một chiều

6.1.2 Mạng
hai chiều



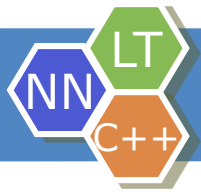
6.1.1 Mảng một chiều

- Khi khai báo mảng, chỉ rõ
 - Tên mảng
 - Kiểu của mảng
 - Bất cứ kiểu dữ liệu nào của C++
 - Số phần tử
- Cú pháp:


```

      <Kieu_dl>   <ten_mang>[ so_phan_tu ];
      int  c[10];      // mảng của 10 số nguyên
      float d[3284];  //mảng của 3284 số thực
      
```
- Khai báo nhiều mảng cùng kiểu
 - Sử dụng dấu phẩy như với các biến bình thường

```
int b[ 100 ], x[ 27 ];
```



6.1.1 Mảng một chiều

- Ví dụ

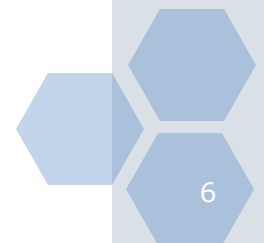
```
#define      SIZE      10

int          a[5];      // mảng a gồm 5 số nguyên

long int     big[100];  // big: chiếm 400 bytes!

double       d[100];    // d: chiếm 800 bytes!

long double  v[SIZE];   // v:10 long doubles
```



6.1.1 Mảng một chiều

- Khai báo và khởi tạo mảng

khởi tạo cho
5 thành phần

```
int      a[5] = { 10, 20, 30, 40, 50 };
double   d[100] = { 1.5, 2.7 };
short    primes[] = { 1, 2, 3, 5, 7, 11, 13 };
long     b[50] = { 0 };
```

Trình biên dịch
xác định kích
thước gồm 7
thành phần

2 thành phần
đầu tiên được
khởi tạo, phần
còn lại: 0

cách nhanh nhất
để khởi tạo tất cả
các thành phần
bằng 0

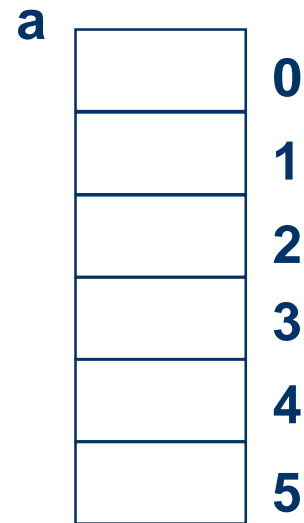
```
int      i = 7;
const int c = 5;
int      a[i];
double   d[c];
short    primes[];
```



6.1.1 Mảng một chiều

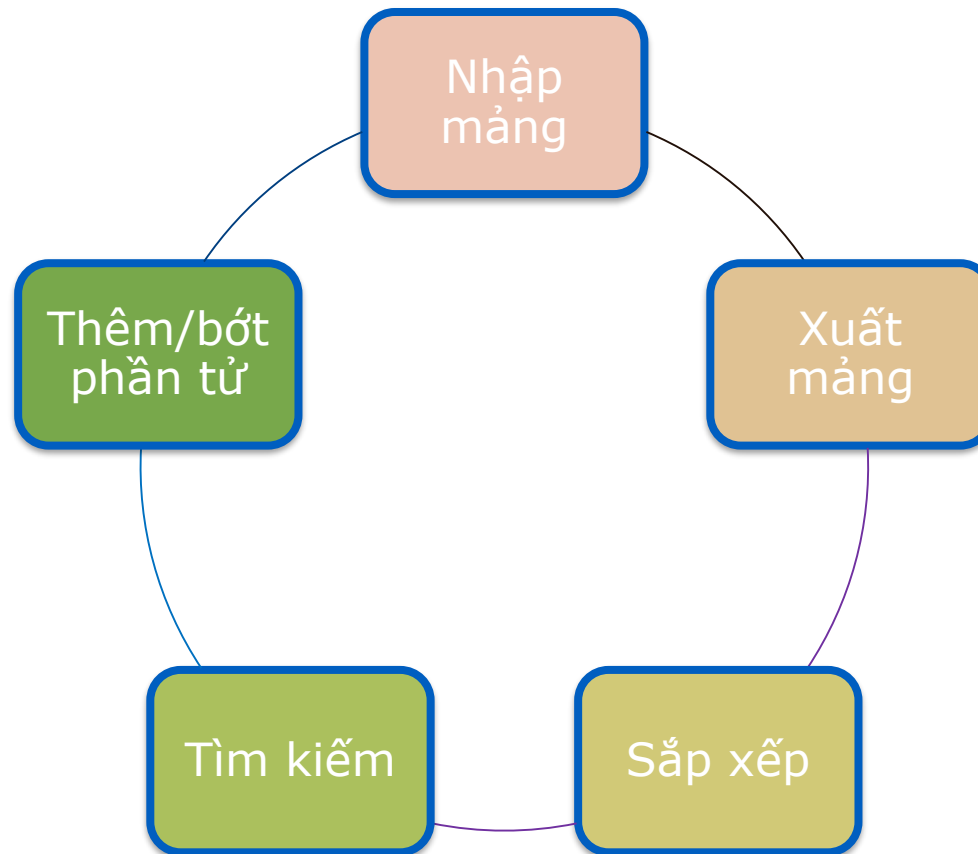
- Truy cập mảng
- ❖ Các thành phần của mảng được truy xuất thông qua chỉ số của chúng $0..n-1$
- ❖ Thao tác truy xuất không kiểm tra giới hạn của chỉ số

```
int main()
{
    int    a[6];
    int    i = 7;
    a[0] = 59;
    a[5] = -10;
    a[i/2] = 2;
    a[6] = 0;
    a[-1] = 5;
    return 0;
}
```



6.1.1 Mảng một chiều

- Các thao tác trên mảng một chiều



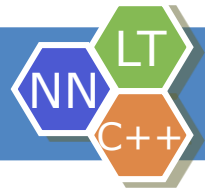
6.1.1 Mảng một chiều

- Nhập mảng

duyệt qua tất
cả các phần tử

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

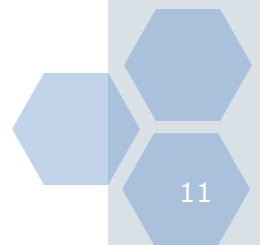
nhập dữ liệu cho a[i]



6.1.1 Mảng một chiều

- Xuất mảng

```
for (int i = 0; i < n; i++)  
    cout<<a[i]<<"\t";
```

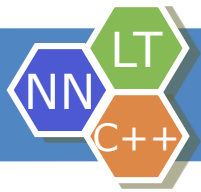


6.1.1 Mảng một chiều

- Tìm kiếm
- ❖ Bài toán: Tìm vị trí x trên mảng a đang có n phần tử.
- ❖ Giải pháp: Tìm tuần tự

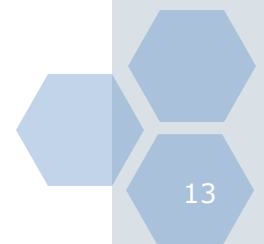
```
//input: dãy (a, n), x
//output: Vị trí của x, -1 nếu không có
int    Search(int a[], int n, int x)
{
    for(int i = 0; i < n; i ++)
        if (a[i] == x)
            return i;
    return -1;
}
```





6.1.1 Mảng một chiều

- Sắp xếp
 - ❖ Bài toán: Sắp xếp các thành phần của (a, n) để thu được dãy tăng dần
 - ❖ Giải pháp: Tìm cách triệt tiêu tất cả các nghịch thế của dãy
 - Thuật toán sắp xếp **Đổi chỗ trực tiếp**



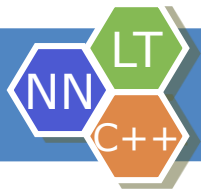
6.1.1 Mảng một chiều

- Sắp xếp

```
void Swap(int &x, int &y)
{
    int tg = x; x = y; y = tg;
}

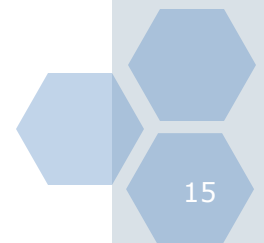
void InterchangeSort(int a[], int n)
{
    int i, j;
    for(i = 0 ; i<n-1 ; i++)
        for(j =i+1; j < n ; j++)
            if(a[j]< a[i])
                Swap(a[i],a[j]);
}
```





6.1.1 Mảng một chiều

- Thêm phần tử
 - ❖ Bài toán: cần thêm thành phần dữ liệu x vào mảng a đang có n thành phần.
 - ❖ Hai trường hợp cần xem xét:
 - Dãy chưa có thứ tự
 - Thêm x vào cuối dãy a .
 - Dãy đã có thứ tự
 - Tìm vị trí thích hợp, chèn x vào



6.1.1 Mảng một chiều

- Thêm phần tử x vào dãy đã sắp xếp tăng dần

```
//input: dãy (a, n) tăng dần, x
//output: dãy (a, n) đã có x ở đúng vị trí

void    Insert(int a[], int n, int x)
{
    int pos = n;

    while ((pos>0) && (a[pos-1]>x))
    {
        a[pos] = a[pos - 1];
        pos --;
    }
    a[pos] = x;
}
```



6.1.1 Mảng một chiều

- Loại bỏ x ra khỏi dãy

```
//input: dãy (a, n), x
//output: dãy (a, n) đã loại bỏ 1 phần tử x

int    Remove(int a[], int n, int x)
{
    int pos = Search(a, n, x);
    if (pos == -1)    //không có x trong dãy
        return 0;
    n--;              //số phần tử giảm đi 1
    for (; (pos < n); pos++)
        a[pos] = a[pos + 1];
    return 1;
}
```



6.1.1 Mảng một chiều

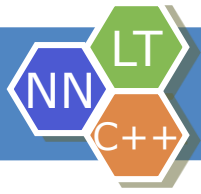
```

1  #include <iostream.h>
2  #include <iomanip.h>
3
4  int main()
5  {
6      int a[10]; // a is an array of 10 integers
7
8      // initialize elements of array a to
9      for ( int i = 0; i < 10; i++ )
10         a[ i ] = 0; // set element at location i to 0
11
12     cout << setw(7)<< "Element" << setw(13) << "Value"<<endl;
13
14     // output contents of array a in tabular format
15     for ( int j = 0; j < 10; j++ )
16         cout << setw(7) << j << setw(13) << a[j] << endl;
17     return 0; // indicates successful termination
18
19 } // end main

```

Khai báo mảng 10 phần tử số nguyên.

Khởi tạo mảng bằng vòng lặp for. Chú ý rằng mảng gồm các phần tử từ a[0] đến a[9].

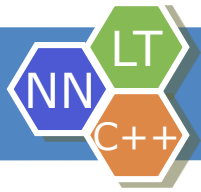


Câu hỏi củng cố bài

1. Điều nào sau đây là khai báo đúng của mảng?
- A. `int arr[10];`
 - B. `int arr;`
 - C. `arr{10};`
 - D. `mang arr[10];`



Multiple Choice



Câu hỏi củng cố bài

2. Lệnh nào sau đây truy cập chính xác phần tử thứ bảy được lưu trữ trong mảng foo gồm 100 phần tử?

A. foo [7];

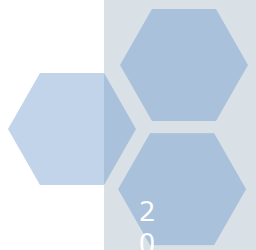
B. foo [6];

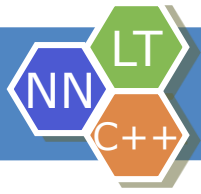
C. foo (7);

D. foo;



Multiple Choice





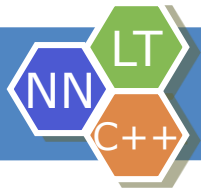
Câu hỏi củng cố bài

3. Cách khai báo mảng nào sau đây là đúng?

- A. `<Kiểu_dl>[]<tên_mảng>;`
- B. `<Kiểu_dl> <tên_mảng>[];`
- C. `<Kiểu_dl> <tên_mảng>[Số phần tử];`
- D. `<Kiểu_dl> <Số_phần_tử>[tên_mảng];`



Multiple Choice



Câu hỏi củng cố bài

4. Chọn lệnh đúng?

A. `int b = {3, 6, 9};`

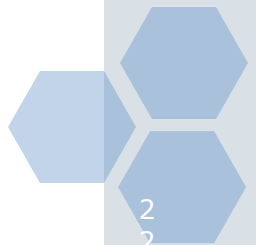
B. `int b[5] = {1,2,3,4}`

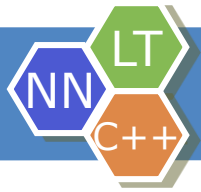
C. `int b[];`

D. `int b[5];`



Multiple Choice





Câu hỏi củng cố bài

5. Đoạn lệnh sau cho kết quả thể nào?

A. In ra n phần tử của mảng 1 chiều

B. Nhập vào các phần tử của mảng n chiều

C. Báo lỗi cú pháp

D. In ra các phần tử của mảng n chiều

```
void A (int arr[],int n) {  
    for(int i = 0; i < n; i++)  
        cout << arr[i];  
}
```



Multiple Choice

Mảng 1 chiều

Khai báo,
truy xuất

Khai báo

```
<Kieu_dl> <ten_mang>[ so_phan_tu ];
<Kieu_dl> <ten_mang>[ so_phan_tu ] = {danh sach gia tri};
```

Truy xuất

ten_mang[chi so]

Nhập,
xuất

Nhập mảng

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

Xuất mảng

```
for(int i = 0; i < n; i++)
    cout<<a[i]<<"\t";
```

Các thao
tác khác

Tìm kiếm

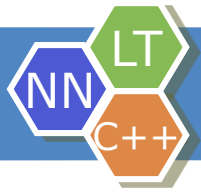
Search

Sắp xếp

Interchange Sort

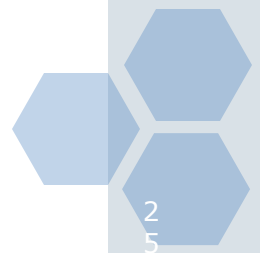
Thêm/bớt pt

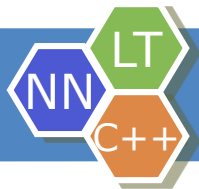
Insert
Remove



Câu hỏi lý thuyết

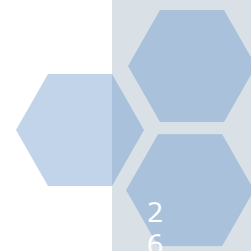
1. Nêu cách khai báo mảng? Cho ví dụ.
2. Nêu giải thuật sắp xếp mảng?
3. Nêu giải thuật tìm kiếm mảng.

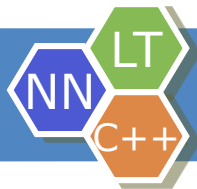




Bài tập

1. Nhập vào một mảng gồm n phần tử nguyên.
Viết chương trình thực hiện các công việc sau:
 - a) Tính trung bình cộng các phần tử âm của mảng
 - b) Tìm và in ra các phần tử chia hết cho 5 trong mảng.
 - c) Sắp xếp mảng theo thứ tự giảm dần





Bài tập

2. Nhập vào một mảng gồm n phần tử nguyên.

Viết chương trình thực hiện các công việc sau:

- a) Tìm, in ra vị trí và giá trị phần tử lớn nhất trong mảng
- b) Tìm và in ra các số hoàn hảo trong mảng
- c) Tìm và in ra các số chính phương trong mảng

