

Chương 4

Kỹ thuật xử lý mảng

Design by Minh An

Email: anvanminh.hau@gmail.com

Nội dung

1. Kỹ thuật xử lý mảng một chiều
2. Kỹ thuật xử lý mảng 2 chiều

Design by Minh An

4.1. Kỹ thuật xử lý mảng một chiều

1. Khái niệm mảng một chiều
2. Sử dụng mảng trong C++
3. Xử lý mảng một chiều trong C++

Design by Minh An

4.1.1. Khái niệm mảng một chiều

- Khái niệm:
 - Mảng một chiều là một dãy liên tiếp các biến có cùng kiểu dữ liệu.
- Khai báo:
`<Kiểu_dữ_liệu> <Tên_mảng>[<Kích_thước>];`
 - Kiểu dữ liệu: có thể là các kiểu cơ sở, cấu trúc
 - Tên mảng: Một tên phù hợp với quy tắc
 - Kích thước: Số nguyên dương

Design by Minh An

Khái niệm mảng một chiều (tt)

• Ví dụ:

```
#define SIZE 10  
int a[SIZE];
```

Mảng a

0	1	2	3	4	5	6	7	8	9

• Các phần tử mảng

- Các phần tử được xác định bởi tên mảng và chỉ số.
- Chỉ số của phần tử bắt đầu từ 0 đến SIZE - 1 với SIZE là kích thước mảng.
- Ký hiệu phần tử: `tên_mảng[chỉ_số]`
- Ví dụ: `a[0]`, `a[1]`, `a[2]`, `a[3]`, ..., `a[9]` lần lượt là các phần tử của mảng a.

Design by Minh An

4.1.2. Sử dụng mảng một chiều

- Mảng một chiều được sử dụng để lưu trữ danh sách.
- Hai mảng có cùng kiểu và cùng kích thước cũng không được xem là tương đương nhau, vì thế không thể gán trực tiếp một mảng cho một mảng khác.
- Không thể gán trị cho toàn bộ mảng, chỉ được gán trị cho từng phần tử của mảng.
- Các thao tác xử lý mảng cần được thiết kế thuật toán chi tiết.

Design by Minh An

4.1.2.1. Đưa dữ liệu vào mảng

- Cách 1: Khởi tạo trong lệnh khai báo

```
int a[SIZE] = {3, -5, 1, 9, 2, 8, 6};
```

- Cách 2: Gán lần lượt từng phần tử

```
void khoiTao(int a[SIZE], int n) {  
    a[0] = 3; a[1] = -5;  
    a[2] = 1; a[3] = 9;  
    a[4] = 2; a[5] = 8;  
    a[6] = 6;  
}
```

Mảng a

0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

Design by Minh An

Đưa dữ liệu vào mảng (tt)

- Cách 3: Gán bằng vòng lặp

```
void khoiTao(int a[SIZE], int n) {  
    for (int i = 0; i < n; i++)  
    {  
        a[i] = i*i;  
    }  
}
```

Mảng a
n = 7

			i						
0	1	2	3	4	5	6	7	8	9
0	1	4	9	16	25	36			

Design by Minh An

Đưa dữ liệu vào mảng (tt)

- Cách 4: Nhập mảng từ bàn phím.

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

Mảng a
n = 7

			i						
0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

Design by Minh An

4.1.2.2. Hiện thị mảng lên màn hình

```
void hienThiMang(int a[SIZE], int n)  
{  
    for (int i = 0; i < n; i++)  
    {  
        cout<<a[i]<<"\t";  
    }  
    cout<<endl;  
}
```

Design by Minh An

4.1.3. Kỹ thuật xử lý mảng một chiều

1. Tìm kiếm
2. Tính toán và thống kê
3. Xóa dữ liệu trong mảng
4. Chèn dữ liệu vào mảng
5. Sắp xếp mảng
6. Ghép mảng, tách mảng

Design by Minh An

4.1.3.1. Kỹ thuật tìm kiếm

Mảng a
n = 7

			i						
0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

```
/* Tìm giá trị lớn nhất trong mảng*/  
int max(int a[SIZE], int n) {  
    int m = a[0];  
    for (int i=1; i < n; i++) {  
        if (m < a[i])  
            m = a[i];  
    }  
    return m;  
}
```

Design by Minh An

4.1.3.2. Tính toán và thống kê số liệu

Mảng a
n=7

0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

/*Tính giá trị trung bình cộng của các phần tử mảng*/

```
float tbc(int a[SIZE],int n) {
    int tong = 0;
    for (int i=0; i<n; i++) {
        tong = tong + a[i];
    }
    return (float)tong/n;
}
```

Design by Minh An

Tính toán và thống kê số liệu (tt)

Mảng a
n=7

0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

```
/*Tính giá trị tb cộng của các phần tử dương*/
void tbcSoDuong(int a[SIZE],int n) {
    int t = 0, d = 0;
    for (int i=0; i<n; i++)
        if (a[i] > 0) {
            t = t + a[i];
            d++;
        }
    if (dem > 0)
        cout<<"\nTBC cac so duong: "<<((float)t/d);
    else cout<<"\nMang khong co so duong nao";
}
```

Design by Minh An

4.1.3.3. Xóa một phần tử dữ liệu trong mảng

- Xóa phần tử thứ $k=3$ trong mảng a đang có n phần tử dữ liệu.

Vị trí thứ k=3

Mảng a
n=7

0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

Mảng a
n=6

0	1	2	3	4	5	6	7	8	9
3	-5	9	2	8	6				

```
void remove(int a[SIZE], int &n, int k) {
    for (int i=k; i<n; i++)
        a[i-1] = a[i];
    n--;
}
```

Design by Minh An

4.1.3.4. Chèn một phần tử dữ liệu vào mảng

- Chèn phần tử $x=12$ vào vị trí thứ $k=3$ trong mảng a.

Vị trí thứ k=3

Mảng a
n=7

0	1	2	3	4	5	6	7	8	9
3	-5	1	9	2	8	6			

x
12

Mảng a
n=6

0	1	2	3	4	5	6	7	8	9
3	-5		1	9	2	8	6		

Mảng a
n=6

0	1	2	3	4	5	6	7	8	9
3	-5	12	1	9	2	8	6		

Design by Minh An

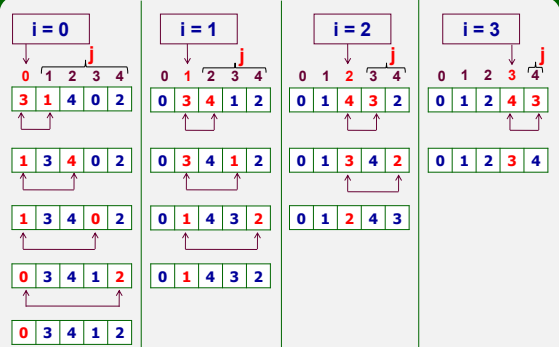
Chèn một phần tử dữ liệu vào mảng (tt)

/*Chèn số x vào vị trí thứ k trong mảng a đang có n phần tử*/

```
void insert(int a[SIZE], n, int k,int x)
{
    for (int i=n-1; i>=k-1; i--)
        a[i+1] = a[i];
    a[k-1] = x;
    n++;
}
```

Design by Minh An

4.1.3.5. Sắp xếp dữ liệu trong mảng



Design by Minh An

Sắp xếp dữ liệu trong mảng (tt)

```
/*Sắp xếp mảng theo chiều tăng dần*/
void sapXep(int a[SIZE], int n) {
    for (int i=0; i<n-1; i++)
        for (int j=i+1; j<n; j++)
            if (a[i] > a[j]) {
                int tg = a[i];
                a[i] = a[j];
                a[j] = tg;
            }
}
```

Design by Minh An

4.1.3.6. Ghép mảng

- Ghép mảng a có 4 phần tử dữ liệu với mảng b có 3 phần tử dữ liệu được mảng c có 7 phần tử dữ liệu

Mảng a					Mảng b				
0	1	2	3	4	0	1	2	3	4
3	-5	0	8		4	2	6		

Mảng c									
0	1	2	3	4	5	6	7	8	9
3	-5	0	8	4	2	6			

Design by Minh An

Ghép mảng (tt)

```
#define N 5
#define M 10
void ghépMang(int a[N], int m, int b[N],
              int n, int c[M], int &k) {
    k = 0;
    for (int i=0; i<m; i++) {
        c[k] = a[i]; k++;
    }
    for (int i=0; i<n; i++) {
        c[k] = b[i]; k++;
    }
}
```

Design by Minh An

4.1.3.7. Tách mảng

- Tách mảng a thành hai mảng b gồm các số dương và c là các số còn lại.

Mảng a									
0	1	2	3	4	5	6	7	8	9
3	-5	6	-8	4	2	0	7		

Mảng b									
0	1	2	3	4	5	6	7	8	9
3	6	4	2	7					

Mảng c									
0	1	2	3	4	5	6	7	8	9
-5	-8	0							

Design by Minh An

Tách mảng (tt)

```
#define M 10
void táchMang(int a[M], int m, int b[M], int &n,
              int c[M], int &k) {
    n = k = 0;
    for (int i=0; i<m; i++) {
        if (a[i]>0) {
            b[n] = a[i]; n++;
        }
        else {
            c[k] = a[i]; k++;
        }
    }
}
```

Design by Minh An

4.1.4. Bài tập

1. Cài đặt chương trình thực hiện các yêu cầu:

- Nhập vào một mảng n số thực ($1 \leq n \leq 30$, n nhập từ bàn phím).
- Hiển thị mảng lên màn hình.
- Tìm và in ra màn hình giá trị nhỏ nhất.
- Tính và in ra màn hình giá trị trung bình cộng của các số âm trong mảng.
- Xóa phần tử thứ k trong mảng (k nhập từ bàn phím), in mảng vừa xóa lên màn hình.
- Sắp xếp mảng theo chiều giảm dần, in mảng vừa sắp ra màn hình.

Design by Minh An