TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Chương 5

Mảng

Khoa công nghệ thông tin
Trường Đại học mở tphcm

1

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Mục tiêu

- Sử dụng cấu trúc mảng để lưu trữ một tập hợp dữ liệu.
- Cách khai báo, khởi tạo và nhập/xuất từng giá trị trong mảng.
- Duyệt, truy cập đến từng giá trị trong mảng.
- Các thao tác cơ bản trên mảng như tìm kiếm, tính toá, thống kê với các giá trị dữ liệu, kiểm tra tính chất mảng, thêm/xóa phần tử trong mảng
- Truyền mảng làm tham số cho hàm.

2

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Nội dung

1. Giới thiệu
2. Khai báo và khởi tạo mảng một chiều
3. Một số thao tác trên mảng
4. Truyền mảng đến hàm

3

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

6.1 Giới thiệu

- Mảng/Dãy (array) là một tập hợp gồm các phần tử có cùng kiểu dữ liệu, được lưu trữ tại các vị trí liên tục trong bộ nhớ.
- Ví dụ:
  - mảng số nguyên
  - mảng số thực
  - mảng ký tự
  - ...

[0]	8
[1]	12
[2]	23
[3]	0
[4]	14
[5]	25
[6]	7
[7]	10
[8]	5
[9]	16

4

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Giới thiệu (tt)

- Mảng một chiều (one-dimensional array) là mảng có các phần tử được sắp xếp theo dạng danh sách (list).

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
list	35	12	27	18	45	16	38	4	17	9	23

- Mảng hai chiều (two-dimensional array) là mảng có các phần tử được sắp xếp theo dạng bảng (table).

inStock	[RED]	[BROWN]	[BLACK]	[GRAY]	[WHITE]
[GM]	8	7	10	6	2
[FORD]	12	10	6	9	17
[TOYOTA]	12	5	10	4	8
[BMW]	8	16	6	3	2
[NISSAN]	14	11	13	9	3
[VOLVO]	7	8	9	4	12

5

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

6.2 Khai báo mảng

- Cú pháp khai báo mảng một chiều:

  dataType arrayName[numberOfElements];

  – dataType: kiểu dữ liệu của mỗi phần tử trong mảng.
  – arrayName: tên của mảng.
  – numberOfElements: số nguyên, cho biết số phần tử (tối đa) của mảng.
- Số phần tử được xác định khi khai báo và không thay đổi.
- Các phần tử được đánh số từ 0 đến numberOfElements - 1

6

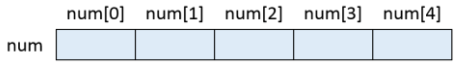
TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Khai báo mảng (tt)

- Ví dụ khai báo mảng tên là num chứa 5 số nguyên:

  int num[5];

  – 5 vị trí liên tục trong bộ nhớ được sử dụng để lưu 5 số nguyên. Nếu kiểu int có kích thước 4 bytes thì mảng sẽ chiếm 20 bytes.



- Phải chỉ rõ số phần tử tối đa của mảng
- Nếu muốn chỉ định số phần tử lưu trong mảng khi chương trình thực thi thì dùng biến con trỏ (pointer) để cấp phát số phần tử của mảng, gọi là mảng cấp phát động (dynamic array).

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Khai báo mảng (tt)

- Khai báo số phần tử của mảng phải là hằng:

```
const int SIZE = 100;
int arrInt[SIZE];
```

  – Ví dụ sau đây sẽ bị lỗi cú pháp:

```
int arrSize;
cout << "Nhap so phan tu cua mang: ";
cin >> arrSize;
int arr[arrSize]; //loi
```

- Định nghĩa kiểu dữ liệu là mảng: dùng **typedef**

```
const int SIZE = 50;
typedef double List[SIZE];
List yourList;//tương tự double yourList[SIZE];
List myList; //tương tự double myList[SIZE];
```

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Khai báo mảng (tt)

- Khai báo và khởi tạo giá trị cho mảng:

dataType arrayName[numberOfElements] = {initialValues};

initialValues: giá trị các phần tử phân cách bằng dấu phẩy (,).

```
//Ví dụ khai báo mảng gồm 3 phần tử và khởi tạo giá trị :
int a[3] = {2, 12, 1};

//tương đương :
int a[3];
a[0] = 2;
a[1] = 12;
a[2] = 1;
```


9

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Khai báo mảng (tt)

- Khi khai báo mảng không chỉ định kích thước nhưng khởi tạo bằng một danh sách các giá trị thì kích thước mảng là tổng số phần tử trong danh sách khởi tạo.

```
double sales[5] = { 12.25, 32.50, 16.90, 23, 45.68 };
//tương đương với:
double sales[] = { 12.25, 32.50, 16.90, 23, 45.68 };
```

- Nếu danh sách khởi tạo có số phần tử ít hơn số phần tử của mảng thì các phần tử còn lại được khởi tạo là 0.

```
//Ví dụ khởi tạo danh sách 10 phần tử có giá trị là 0.
int list[10] = {0};
//Ví dụ chỉ khởi tạo giá trị cho 3 phần tử đầu, còn lại là 0.
int list[10] = {8, 5, 12};
//tương đương với
int list[10] = { 8, 5, 12, 0, 0, 0, 0, 0, 0, 0 };
```


10

Gán giá trị cho phần tử trong mảng

- Cú pháp:

  arrayName[index] = expression;

  – index: chỉ số (vị trí) là số nguyên hoặc biểu thức có giá trị là số nguyên. Nếu mảng có số phần tử là MAXSIZE thì index có giá trị từ 0 đến MAXSIZE - 1.

  – expression: biểu thức có cùng kiểu với kiểu dữ liệu của mảng.

Gán giá trị cho phần tử trong mảng

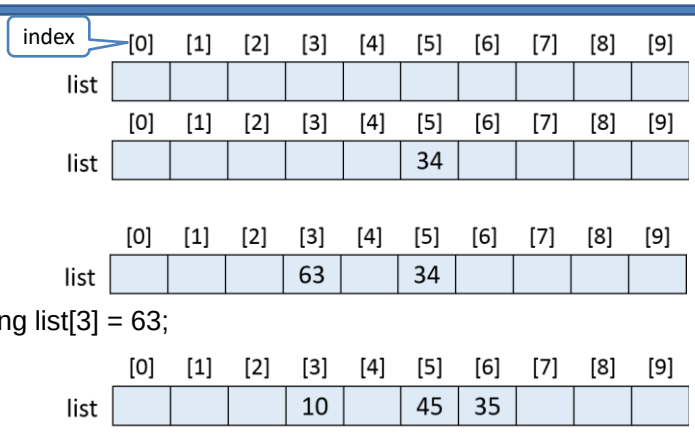
– Ví dụ:

```
int list[10];

list[5] = 34;

int i = 3;
list[i] = 63;
//tuong duong list[3] = 63;

list[3] = 10;
list[6] = 35;
list[5] = list[3] + list[6];
```



TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

6.3 Một số thao tác trên mảng

- Các thao tác thường gặp trên mảng gồm:
  - Nhập và in dữ liệu
  - Tìm một giá trị, tìm giá trị nhỏ nhất, lớn nhất
  - Thống kê: đếm, tính tổng, giá trị trung bình,…
  - Thêm/Xóa
  - Sắp xếp...
- Các thao tác này thường dùng vòng lặp để truy cập phần tử trong mảng.

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Nhập mảng bằng cách gán trực tiếp:

```
//Chương trình gán các giá trị = 1 vào 10 phần
tử mảng, sau đó in ra mảng
#include <iostream>
using namespace std;
int main()
{
    const int SIZE = 10;
    double a[SIZE];
    for (int i = 0; i < SIZE; i++)
        a[i] = 1;
    for (int i = 0; i < SIZE; i++)
        cout << a[i] << " ";
    cout << endl;
    return 0;
}
```

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Nhập lần lượt từng giá trị vào mảng:

```
//Chương trình nhập mảng gồm 10 số, sau đó in ra mảng.
#include <iostream>
using namespace std;
int main()
{
    const int MSIZE = 10;
    double a[SIZE];
    cout << "Nhap " << SIZE << " so double: ";
    for (int i = 0; i < SIZE; i++)
        cin >> a[i];
    for (int i = 0; i < SIZE; i++)
        cout << a[i] << " ";
    cout << endl;
    return 0;
}
```


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Chương trình tính tổng và trung bình cộng các phần tử trong mảng.

```
int main()
{
    double a[] = { 8, 6, 7, 1, 4, 2, 3, 10, 9, 5 };
    int n = sizeof(a) / sizeof(a[0]);
    cout << "Cac gia tri trong mang:" << endl;
    for (int i = 0; i < n; i++)
        cout << a[i] << " ";
    double tong = 0;
    for (int i = 0; i < n; i++)
        tong += a[i];
    cout << "\nTong = " << tong;
    cout << "\nTrung binh = " << tong / n;
    return 0;
}
```

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Tìm kiếm vị trí một phần tử trong mảng
  - Xuất phát từ đầu mảng, bắt đầu từ vị trí 0
  - So sánh giá trị phần tử đang xét và giá trị muốn tìm
  - Nếu thấy: trả về vị trí hiện tại, dừng
  - Không thấy: xét phần tử tiếp theo
  - Nếu xét đến phần tử cuối cùng vẫn không có phần tử nào bằng giá trị muốn tìm thì dừng
  - Ví dụ tìm 27 trong mảng gồm có 7 số nguyên:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
a	35	12	27	18	45	16	38

- Giải pháp: Tìm từ đầu cho đến cuối mảng:
  So sánh 27 với a[0], a[1], a[2], ..., a[6].
  Do a[2] == 27 nên quá trình tìm kiếm thành công, trả về vị trí 2.

17

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Tìm kiếm vị trí một phần tử trong mảng

```
int main()
{
    int a[] = { 35, 12, 27, 18, 45, 16, 38 };
    int n = sizeof(a) / sizeof(a[0]);
    int x, vitri;
    cout << "Nhap so can tim: ";
    cin >> x;
    vitri = 0;
    while (vitri < n && a[vitri] != x)
        vitri++;
    if (vitri < n)
        cout << "Tim thay tai vi tri " << vitri << endl;
    else
        cout << "Khong tim thay" << endl;
    return 0;
}
```


18

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Tìm phần tử lớn nhất
  - Dùng một biến (max) lưu giá trị phần tử đầu tiên
  - Lần lượt so sánh giá trị của max và các phần tử sau nó
  - Nếu giá trị phần tử đang xét > max: gán max bằng phần tử đó
  - Xét phần tử kế tiếp cho đến hết phần tử cuối
  - Giá trị biến max chính là giá trị lớn nhất trong mảng

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Ví dụ tìm giá trị lớn nhất trong mảng sau đây:

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
a	12.5	8.35	19.60	25.00	14.00	39.43	35.90	98.23	66.65	35.64

- Giải pháp:
  - max = a[0]
  - So sánh max với a[1],..., a[n-1] và cập nhật lại max

i	max	a[i]	max < a[i]?
1	12.50	8.35	false
2	12.50	19.60	true
3	19.60	25.00	true
...	...	...	...
7	39.43	98.23	true
8	98.23	66.65	false
9	98.23	35.64	false

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số thao tác trên mảng

- Tìm giá trị lớn nhất trong mảng

```
int main()
{
    double a[] = {12.5, 8.35, 19.6, 25, 14, 39.43,
                  35.9, 98.23, 66.65, 35.64};
    int n = sizeof(a) / sizeof(a[0]);
    double max = a[0];
    for (int i = 1; i <= n - 1; i++)
        if (max < a[i])
            max = a[i];

    cout << "Gia tri lon nhat = " << max << endl;

    return 0;
}
```


21

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số lưu ý

- Không truy xuất phần tử ngoài phạm vi mảng.
- Ví dụ:

```
int a[10];
for (int i = 0; i <= 10; i++)
    a[i] = 1;
```

  - các chỉ số i hợp lệ là 0, 1, 2,..., 9
  - khi i = 10, vòng lặp kiểm tra điều kiện i <= 10 đúng và gán 0 vào a[10], nhưng a[10] không tồn tại!
- Không dùng phép gán một mảng cho một mảng khác mà dùng vòng lặp để sao chép các giá trị của mảng

```
int a[5] = {0, 4, 8, 12, 16};
int b[5];
b = a; //illegal
```

```
for (int i = 0; i < 5; i++)
    b[i] = a[i];
```


22

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Một số lưu ý

- Không dùng cin/cout với biến mảng.
  - Ví dụ:

```
int a[10]; int b[10];
cin >> a;
//...//illegal
cout << b;//illegal
```

- Dùng vòng lặp để nhập/xuất dữ liệu trong mảng:

```
for (int i = 0; i < 10; i++)
    cin >> a[i];
```

- Không dùng phép so sánh đối với biến mảng:

```
if (a < b)//illegal
    //...
```


TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

6.4 Truyền mảng cho hàm

- Trong C++, biến mảng luôn được truyền cho hàm bằng tham chiếu (pass-by-reference).
  - Không cần dùng ký hiệu & khi khai báo tham số là mảng.
- Khi tham số của hàm là mảng một chiều:
  - không cần khai báo số phần tử tối đa của mảng
  - Nhưng phải có tham số cho biết số phần tử hiện có trong mảng.
- Ví dụ hàm khởi tạo các giá trị ban đầu cho mảng:

```
void khoitao(int arr[], int arrSize)
{
    for (int i = 0; i < arrSize; i++)
        arr[i] = 0;
}
```

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Truyền mảng cho hàm

- Khi truyền tham chiếu, hàm có thể làm thay đổi giá trị của tham số.
- Muốn ngăn không cho hàm thay thay đổi giá trị của tham số mảng, ta dùng tham chiếu hằng.

```
void xuat(const int arr[], int arrSize)
{
    for (int i = 0; i < arrSize; i++)
        cout << arr[i] << " ";
    cout << endl;
}
```

- C++ không cho phép hàm trả về giá trị có kiểu mảng.
- Ví dụ hàm tính giá trị trung bình của một mảng số thực.

25

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Truyền mảng cho hàm

```
//tính giá trị trung bình của một mảng số nguyên
#include <iostream>
using namespace std;
//khai báo nguyên mẫu hàm
double tinhTrungbinh(double arr[], int arrSize);
int main()
{
    int a[] = {1,2,3,4,5,6,7,8,9,10};
    int n = sizeof(a) / sizeof(a[0]);
    cout << "Trung binh = " << tinhTrungbinh(a, n) << endl;
    return 0;
}
double tinhTrungbinh(double arr[], int arrSize)
{
    double tong = 0.0;
    for (int i = 0; i < arrSize; i++)
        tong += arr[i];
    return tong / arrSize;
}
```


26

TRƯỜNG ĐẠI HỌC MỞ TP. HCM
Cơ hội học tập cho mọi người

Q & A