

MẢNG

Kỹ thuật Lập trình (CO1027)

Ngày 6 tháng 3 năm 2021

ThS. Trần Ngọc Bảo Duy

Khoa Khoa học và Kỹ thuật Máy tính

Trường Đại học Bách Khoa, ĐHQG-HCM

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Tổng quan

① Mảng

Sự cần thiết của mảng
Định nghĩa

② Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

③ Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

Sự cần thiết của mảng

Array

ThS.
Trần Ngọc Bảo Duy



Bài toán

Một chương trình quản lý bán hàng cần lưu trữ tạm N món hàng trong bộ nhớ, nếu ta chỉ sử dụng các kiểu dữ liệu cơ bản (primitive type):

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều



Bài toán

Một chương trình quản lý bán hàng cần lưu trữ tạm N món hàng trong bộ nhớ, nếu ta chỉ sử dụng các kiểu dữ liệu cơ bản (primitive type):

- Phải cần đến $N \times M$ biến:
 - M là số thành phần dữ liệu của một món hàng.
 - Ví dụ: $N = 100$ món hàng, $M = 10$ thành phần dữ liệu, ta cần $N \times M = 1000$ biến.

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều



Bài toán

Một chương trình quản lý bán hàng cần lưu trữ tạm N món hàng trong bộ nhớ, nếu ta chỉ sử dụng các kiểu dữ liệu cơ bản (primitive type):

- Phải cần đến $N \times M$ biến:
 - M là số thành phần dữ liệu của một món hàng.
 - Ví dụ: $N = 100$ món hàng, $M = 10$ thành phần dữ liệu, ta cần $N \times M = 1000$ biến.
- Nhận định: Khả thi nhưng không hợp lý. Chương trình khó đọc và khó phát triển.

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Sự cần thiết của mảng

Giải pháp

- ❶ Gom các dữ liệu để mô tả một món hàng cùng với nhau, ta có `struct` hỗ trợ.
- ❷ Lưu trữ N món hàng dùng kiểu dữ liệu mảng (nâng cao hơn là danh sách).

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Sự cần thiết của mảng

Array

ThS.
Trần Ngọc Bảo Duy



Giải pháp

- ❶ Gom các dữ liệu để mô tả một món hàng cùng với nhau, ta có `struct` hỗ trợ.
- ❷ Lưu trữ N món hàng dùng kiểu dữ liệu mảng (nâng cao hơn là danh sách).

Về kỹ thuật, C/C++ hỗ trợ:

- Mảng (array) để lưu trữ liên tục các phần tử có cùng một kiểu dữ liệu.
- Con trỏ (pointer) để từ đó phát triển các loại danh sách nếu muốn.

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Định nghĩa

Định nghĩa

Mảng là một dãy các phần tử **cùng một kiểu**, **nằm liền kề nhau** trong bộ nhớ.

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Định nghĩa

Định nghĩa

Mảng là một dãy các phần tử **cùng một kiểu**, **nằm liền kề nhau** trong bộ nhớ.

Ví dụ:

0	1	2	3	4	5
10	20	30	40	50	60

Hình: Một mảng gồm 6 phần tử số nguyên

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

0	1	2	3	4	5
10	20	30	40	50	60

Hình: Một mảng gồm 6 phần tử số nguyên

- Sáu số này nằm liên tục nhau trong bộ nhớ: Nếu ô nhớ đầu tiên, chứa giá trị 10, bắt đầu ở BYTE có địa chỉ **100** trong vùng nhớ của chương trình
 - Địa chỉ của ô nhớ chứa 20: **104**
 - Địa chỉ của ô nhớ chứa 30: **108**
 - Địa chỉ của ô nhớ chứa 40: **112**
 - Địa chỉ của ô nhớ chứa 50: **116**
 - Địa chỉ của ô nhớ chứa 60: **120**



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

0	1	2	3	4	5
10	20	30	40	50	60

Hình: Một mảng gồm 6 phần tử số nguyên

- Các phần tử trong mảng được đánh **chỉ số** để truy xuất.

- Phần tử đầu tiên **LUÔN LUÔN CÓ** chỉ số là 0.
- Các phần tử kế tiếp theo là 1, 2, ...

Vì vậy, chỉ số của các phần tử:

- Ô nhớ chứa 10 có chỉ số là 0
- Ô nhớ chứa 20 có chỉ số là 1
- Ô nhớ chứa 30 có chỉ số là 2
- Ô nhớ chứa 40 có chỉ số là 3
- Ô nhớ chứa 50 có chỉ số là 4
- Ô nhớ chứa 60 có chỉ số là 5



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

0	1	2	3	4	5
10	20	30	40	50	60

Hình: Một mảng gồm 6 phần tử số nguyên

- Tổng quát: Một mảng có **N** phần tử thì chỉ số các phần tử là 0, 1, ... và cuối cùng là **(N - 1)** - không phải **N**



Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Địa chỉ của các ô nhớ trong mảng

Array

ThS.
Trần Ngọc Bảo Duy



Về lý thuyết, để tính địa chỉ (đ/c) của ô nhớ có chỉ số **k**, chương trình dùng công thức sau:

$$\text{Địa chỉ} = \text{đ/c phần tử đầu tiên} + k \times (\text{kích thước phần tử})$$

Do đó, chương trình dễ dàng chỉ ra ngay một phần tử có chỉ số bất kỳ, ta gọi đó là **TRUY CẬP NGẪU NHIÊN**.

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Địa chỉ của các ô nhớ trong mảng

Array

ThS.
Trần Ngọc Bảo Duy



Trong C/C++, trình biên dịch đã biết trước kích thước của phần tử trong mảng. Do đó, công thức lấy địa chỉ của phần tử có chỉ số **k** là

$$\text{Địa chỉ của phần tử có chỉ số } k = \mathbf{first} + k$$

trong đó, **first** là địa chỉ của phần tử đầu tiên của mảng, có thể lấy thông qua tên của biến mảng.

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

MẢNG MỘT CHIỀU

Khai báo mảng 1 chiều

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo

Đọc và ghi trên mảng hai chiều

Để khai báo một mảng một chiều trong C/C++, ta sử dụng cú pháp:

```
<datatype> <array_name>[<size>;
```

trong đó:

- <datatype> là kiểu dữ liệu của các phần tử trong mảng
- <array_name> là tên mảng, do người lập trình đặt.
- <size> là một hằng thể hiện kích thước của mảng.

Ví dụ

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a[6];
6      int b[6] = {10, 20, 30};
7      int c[6] = {10, 20, 30, 40, 50, 60};
8
9      return 0;
10 }
```

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

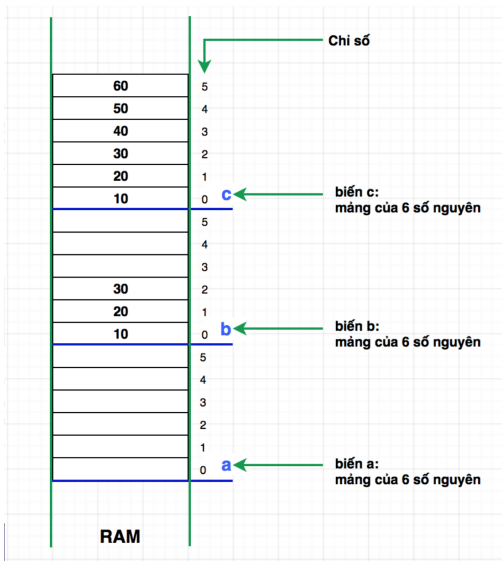
Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Ví dụ



Hình: Hình ảnh trong bộ nhớ của các mảng a, b, và c

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Số phần tử của mảng

Số phần tử của mảng phải biết trước thời điểm biên dịch, do đó phải là một hằng số không âm:

- Sử dụng macro: `#define MAX_SIZE`
- Khai báo hằng số nguyên: `const int MAX_SIZE`

```
1  #include <iostream>
2  using namespace std;
3
4  #define MAX_SIZE 6
5
6  int main() {
7      const int max_size = 10;
8      int a[MAX_SIZE];
9      int b[max_size];
10     return 0;
11 }
```



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Số phần tử của mảng

Số phần tử của mảng phải biết trước thời điểm biên dịch, do đó phải là một hằng số không âm:

- Sử dụng macro: `#define MAX_SIZE`
- Khai báo hằng số nguyên: `const int MAX_SIZE`

```
1  #include <iostream>
2  using namespace std;
3
4  #define MAX_SIZE 6
5
6  int main() {
7      const int max_size = 10;
8      int a[MAX_SIZE];
9      int b[max_size];
10     return 0;
11 }
```

Một số trình biên dịch cho phép khai báo kích thước với biến dữ liệu, tuy nhiên, đó không phải là quy tắc chuẩn được định nghĩa bởi ANSI cho C++ (<http://www.gotw.ca/iso/>).



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Đọc và ghi trên mảng 1 chiều

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

Phương pháp

Có hai cách để đọc và ghi trên mảng:

- ❶ Dùng chỉ số để lấy phần tử quan tâm.
- ❷ Tự tính toán địa chỉ và lấy phần tử quan tâm.

Dùng chỉ số

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int c[6] = {10, 20, 30, 40, 50, 60};
6      int id = 0;
7      /*Write to element*/
8      c[3] = 99;
9      c[id + 1] = 100;
10     /*Read and print element*/
11     cout << "c[3]_=_ " << c[3] << endl;
12     cout << "c[" << id + 1 << "]_=_ "
13           << c[id + 1] << endl;
14     return 0;
15 }
```

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Dùng chỉ số

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int c[6] = {10, 20, 30, 40, 50, 60};
6      int id = 0;
7      /*Write to element*/
8      c[3] = 99;
9      c[id + 1] = 100;
10     /*Read and print element*/
11     cout << "c[3]_=_ " << c[3] << endl;
12     cout << "c[" << id + 1 << "]_=_ "
13           << c[id + 1] << endl;
14     return 0;
15 }
```

Chương trình trên sẽ in ra đoạn kết quả sau:

```
c[3] = 99
c[1] = 100
```

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Tự tính toán địa chỉ và lấy phần tử quan tâm

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int c[6] = {10, 20, 30, 40, 50, 60};
6      int id = 0;
7      /*Write to element*/
8      *(c + 3) = 99;
9      *(c + (id + 1)) = 100;
10     /*Read and print element*/
11     cout << "c[3]_=_ " << *(c + 3) << endl;
12     cout << "c[" << id + 1 << "]_=_ "
13           << *(c + (id + 1)) << endl;
14     return 0;
15 }
```


Tự tính toán địa chỉ và lấy phần tử quan tâm

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int c[6] = {10, 20, 30, 40, 50, 60};
6      int id = 0;
7      /*Write to element*/
8      *(c + 3) = 99;
9      *(c + (id + 1)) = 100;
10     /*Read and print element*/
11     cout << "c[3] = " << *(c + 3) << endl;
12     cout << "c[" << id + 1 << "] = "
13           << *(c + (id + 1)) << endl;
14     return 0;
15 }
```

Chương trình trên sẽ in ra đoạn kết quả sau:

```
c[3] = 99
c[1] = 100
```

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Tự tính toán địa chỉ và lấy phần tử quan tâm

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int c[6] = {10, 20, 30, 40, 50, 60};
6      int id = 0;
7      /*Write to element*/
8      *(c + 3) = 99;
9      *(c + (id + 1)) = 100;
10     /*Read and print element*/
11     cout << "c[3]_=_ " << *(c + 3) << endl;
12     cout << "c[" << id + 1 << "]"_=_ "
13         << *(c + (id + 1)) << endl;
14     return 0;
15 }
```

Các bước:

- ❶ Tính toán địa chỉ theo công thức $\text{first} + k$: $c + 3$
- ❷ Lấy địa chỉ tại một địa chỉ thông qua toán tử *: $*(c + 3)$

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Tự tính toán địa chỉ và lấy phần tử quan tâm

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int c[6] = {10, 20, 30, 40, 50, 60};
6      int id = 0;
7      /*Write to element*/
8      *(c + 3) = 99;
9      *(c + (id + 1)) = 100;
10     /*Read and print element*/
11     cout << "c[3]_=_ " << *(c + 3) << endl;
12     cout << "c[" << id + 1 << "]"_=_ "
13           << *(c + (id + 1)) << endl;
14     return 0;
15 }
```

Như vậy first hay địa chỉ phần tử đầu tiên của mảng, có thể được lấy thông qua hai cách:

- 1 Tên mảng: như trên, dùng tên c
- 2 Thông qua truy cập chỉ số: &c[0], với toán tử & là toán tử lấy địa chỉ.



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Một số kỹ thuật trên mảng một chiều

- 1 Duyệt qua tất cả các phần trên mảng
- 2 Tính các giá trị thống kê từ mảng: tổng, lớn nhất, nhỏ nhất, trung bình, trung vị, độ lệch chuẩn, ...
- 3 Xử lý từng phần tử trên mảng bằng một giải thuật nào đó.
- 4 Hoán đổi từng cặp phần tử trên mảng.
- 5 Sắp xếp các phần tử trên mảng.
- 6 Tìm kiếm một phần tử trên mảng: tìm kiếm tuần tự, tìm kiếm nhị phân.

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo

Đọc và ghi trên mảng hai chiều



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

MẢNG HAI CHIỀU

- Các ma trận trong toán học (Đại số tuyến tính) là những mảng hai chiều.
- Ảnh được lưu trữ trong máy tính là mảng hai chiều các điểm ảnh (pixel).
- Đồ thị (mạng của các đối tượng) có thể được biểu diễn dùng mảng 2 chiều.



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

Mô hình và Lưu trữ vật lý

10	20	30	40
50	60	70	80
90	100	110	120

Hình: Mô hình một mảng hai chiều: 3 hàng \times 4 cột

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

Mô hình và Lưu trữ vật lý

10	20	30	40
50	60	70	80
90	100	110	120

Hình: Mô hình một mảng hai chiều: 3 hàng \times 4 cột

0	1	2	3	4	5	6	7	8	9	10	11
10	20	30	40	50	60	70	80	90	100	110	120

Hình: Lưu trữ của mảng 2 chiều: tuyến tính hoá mảng 2 chiều

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo

Đọc và ghi trên mảng hai chiều

Cách lưu trữ mảng 2 chiều

Array

ThS.
Trần Ngọc Bảo Duy



Phương pháp lưu trữ

- Các phần tử được lưu trữ liên tục, hết hàng đến hàng:
Nếu phần tử đầu tiên (giá trị 10) bắt đầu ở BYTE có địa chỉ là **100**
 - Địa chỉ của ô nhớ chứa 20: **104**
 - Địa chỉ của ô nhớ chứa 30: **108**
 - Địa chỉ của ô nhớ chứa 40: **112**
 - Địa chỉ của ô nhớ chứa 50: **116**
 - Địa chỉ của ô nhớ chứa 60: **120**
 - Địa chỉ của ô nhớ chứa 90: **132**
 - Địa chỉ của ô nhớ chứa 100: **136**
 - ...

Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo
Đọc và ghi trên mảng hai chiều

Cách lưu trữ mảng 2 chiều

Array

ThS.
Trần Ngọc Bảo Duy



Phương pháp lưu trữ

- Các phần tử trong mảng 2 chiều được đánh chỉ số để truy cập, dùng 2 chỉ số:
 - Chỉ số hàng bắt đầu từ 0 đến (Số hàng - 1)
 - Chỉ số cột bắt đầu từ 0 đến (Số cột - 1)

	0	1	2	3
0	10	20	30	40
1	50	60	70	80
2	90	100	110	120

Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Cách lưu trữ mảng 2 chiều

Array

ThS.
Trần Ngọc Bảo Duy



Phương pháp lưu trữ

- Chương trình có thể tính địa chỉ của ô nhớ bắt đầu của phần tử có chỉ số [row, col] dễ dàng:

Đ/c phần tử [row, col] = first + [row * COLS + col]

trong đó:

- first: địa chỉ của phần tử đầu tiên, là tên mảng
- COLS: số phần tử trên mỗi hàng

Mảng

Sự cần thiết của mảng

Định nghĩa

Mảng một chiều

Khai báo

Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Khai báo mảng hai chiều

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a[3][4];
6      int b[3][4] = { {10, 20, 30}      };
7      int c[3][4] = { {10, 20, 30},
8                      {50, 60}
9                      };
10     int d[3][4] = { {10, 20, 30, 40}
11                    {50, 60, 70, 80},
12                    {90, 100, 110, 120}
13                    };
14     return 0;
15 }
```

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng

Khai báo

Đọc và ghi trên mảng hai chiều

Đọc và ghi trên mảng hai chiều

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a[3][4];
6      int r = 0;
7      int c = 2;
8
9      a[r][c] = 99;
10
11     cout << "a[" << r << "][" << c << "]="
12         << a[r][c] << endl;
13     return 0;
14 }
```

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo

Đọc và ghi trên mảng hai chiều

Đọc và ghi trên mảng hai chiều

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a[3][4];
6      int r = 0;
7      int c = 2;
8
9      a[r][c] = 99;
10
11     cout << "a[" << r << "][" << c << "]="
12         << a[r][c] << endl;
13     return 0;
14 }
```

Đoạn chương trình trên in ra :

a[0][2] = 99

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo

Đọc và ghi trên mảng hai chiều

Một số kỹ thuật trên mảng hai chiều

Array

ThS.
Trần Ngọc Bảo Duy



Mảng

Sự cần thiết của mảng
Định nghĩa

Mảng một chiều

Khai báo
Đọc và ghi trên mảng 1 chiều

Mảng hai chiều

Ứng dụng
Khai báo

Đọc và ghi trên mảng hai chiều

- Duyệt qua từng phần tử trong mảng
- Duyệt qua từng phần tử trong cùng một hàng
- Duyệt qua từng phần tử trong cùng một cột
- Với ma trận vuông:
 - Duyệt các phần tử trên đường chéo chính
 - Duyệt các phần tử trên đường chéo phụ