**Mảng 1 Chiều**

**1.Định Nghĩa Và Tính Chất**

Một chương trình thường được cấu thành từ 2 phần là cấu trúc dữ liệu và thuật toán. Cấu trúc dữ liệu là nơi bạn sẽ lưu trữ dữ liệu của bài toán, ví dụ như 1 dãy số hay 1 chuỗi các ký tự.

Định nghĩa : Mảng 1 chiều là cấu trúc dữ liệu (data structure) dùng để lưu trữ nhiều phần tử có cùng kiểu dữ liệu, các phần tử trong mảng được lưu trữ tại các ô nhớ cạnh nhau trong bộ nhớ

Tính chất :

* Đơn giản và dễ sử dụng
* Thuận tiện trong việc truy xuất giá trị của các phần tử
* Các phần tử trong mảng được gán chỉ số để thuận tiện trong việc quản lý
* Là cấu trúc dữ liệu được sử dụng nhiều nhất
* Kích thước cố định, không thể mở rộng và thu hẹp

**2.Khai Báo Mảng**

Cú pháp : Kiểu\_Dữ\_Liệu Tên\_mảng[Số\_Lượng\_Phần\_Tử];

Khi khai báo mảng thì mảng sẽ chiếm số byte = số lượng phần tử x kích thước của 1 phần tử.

Lưu ý khi các bạn khai báo mảng như trên thì các phần tử trong mảng thường là các giá trị không xác định, các giá trị này là ngẫu nhiên.

Ví dụ :

|  |  |
| --- | --- |
| Khai báo | Ý nghĩa |
| int a[100]; | Khai báo mảng chứa các số int tối đa 100 phần tử |
| float b[1000]; | Khai báo mảng chứa các số float tối đa 1000 phần tử |
| double c[200]; | Khai báo mảng chứa các số double tối đa 200 phần tử |
| char c[20000]; | Khai báo mảng chứa các ký tự tối đa 20000 phần tử |

Bạn có thể vừa khai báo vừa khởi tạo giá trị cho các phần tử trong mảng theo cách sau :

#include <stdio.h>

int main(){

//các phần tử của a có giá trị ngẫu nhiên

int a[100];

//các phần tử trong b lần lượt là 2 8 0 4 2 5

int b[6] = {2, 8, 0, 4, 2, 5};

return 0;

}

Trong trường hợp mảng có nhiều phần tử nhưng bạn chỉ khởi tạo giá trị cho 1 vài phần tử đầu tiên thì các phần tử không được khởi tạo sẽ có giá trị là 0.

#include <stdio.h>

int main(){

//các phần tử của a có giá trị ngẫu nhiên

int a[100];

//các phần tử trong b lần lượt là 2 8 0 4 2 5 0 0 0 0 0 .....

int b[100] = {2, 8, 0, 4, 2, 5};

return 0;

}

**3. Các Thao Tác Trên Mảng**

**Duyệt mảng : ﻿**

Các phần tử trong mảng được đánh chỉ số tính từ 0, ví dụ mảng của bạn có 100 phần tử thì chỉ số sẽ được đánh số từ 0 tới 99. Tổng quát nếu mảng có N phần tử thì chỉ số sẽ được đánh từ 0 tới N - 1

Để truy cập các phần tử trong mảng thông qua chỉ số ta dùng cú pháp : tên\_mảng[chỉ\_số];

Thông qua chỉ số bạn có thể truy xuất, thay đổi.. giá trị của phần tử trong mảng tại chỉ số tương ứng.

Ví dụ với mảng b[6] = {2, 8, 0, 4, 2, 5} thì b[3] sẽ có giá trị là 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **b** | 2 | 8 | 0 | 4 | 2 | 5 |
| **Chỉ số** | 0 | 1 | 2 | 3 | 4 | 5 |

Để duyệt qua các phần tử trong mảng thì bạn duyệt qua từng chỉ số :

#include <stdio.h>

int main(){

int n = 10;

int a[10] = {3, 2, 1, 4, 5, 8, 9, 7, 6, 10};

for(int i = 0; i < n; i++){

printf("%d ", a[i]);

}

printf("\n");

for(int i = n - 1; i >= 0; i--){

printf("%d ", a[i]);

}

return 0;

}

Output :

3 2 1 4 5 8 9 7 6 10

10 6 7 9 8 5 4 1 2 3

**Thay đổi giá trị trong mảng**

#include <stdio.h>

int main(){

int b[6] = {1, 2, 3, 4, 5, 6};

b[0] = 28;

b[1] = 100;

b[2] = 14;

b[3] += 10;

b[4] \*= 2;

b[5] /= 2;

for(int i = 0; i < 6; i++){

printf("%d ", b[i]);

}

return 0;

}

Output :

28 100 14 14 10 3

**Nhập mảng từ bàn phím :**

Để nhập mảng từ bàn phím bạn sẽ nhập số lượng phần tử trong mảng trước sau đó nhập từng phần tử trong mảng từ đầu tới cuối

Khi nhập mảng từ bàn phím bạn có thể khai báo mảng có kích thước cố định (1000, 2000, 100...) hoặc khai báo dựa trên số lượng phần tử mà đề bài cho.

Code 1 : Nhập mảng  từ bàn phím với mảng có kích thước cố định

#include <stdio.h>

int main(){

//Lưu ý là mảng a chỉ lưu được tối đa 1000 phần tử

int n, a[1000];

printf("Nhap n : ");

scanf("%d", &n);

for(int i = 0; i < n; i++){

printf("Nhap a[%d] : ", i);

scanf("%d", &a[i]);

}

printf("Mang vua nhap : ");

for(int i = 0; i < n; i++){

printf("%d ", a[i]);

}

return 0;

}

Chú ý : Khi bạn sử dụng mảng cố định thì cần lưu ý tới dữ liệu của đề bài, ví dụ trong code trên mình khai báo mảng a[1000] thì nó sẽ đúng nếu bạn sử dụng mảng a với các bài toán có số lượng phần tử ≤ 1000. Trường hợp đề bài cho nhiều hơn 1000 phần tử thì sẽ bị sai vì không đủ bộ nhớ.

Code 2 : Nhập mảng  từ bàn phím với mảng có kích thước vừa đủ

#include <stdio.h>

int main(){

//N là số lượng phần tử của mảng

int n;

printf("Nhap n : ");

int a[n]; //Khai báo mảng a vừa đủ n phần tử

scanf("%d", &n);

for(int i = 0; i < n; i++){

printf("Nhap a[%d] : ", i);

scanf("%d", &a[i]);

}

printf("Mang vua nhap : ");

for(int i = 0; i < n; i++){

printf("%d ", a[i]);

}

return 0;

}

**4. Mảng Làm Tham Số**

Bạn có thể sử dụng mảng làm tham số cho hàm, thông thường khi hàm làm tham số cho hàm bạn nên cung cấp thêm 1 tham số đi kèm là số lượng phần tử trong mảng.

Ví dụ 1 : Hàm nhập mảng và in mảng

#include <stdio.h>

void nhap(int a[], int n){

for(int i = 0; i < n; i++){

printf("Nhap a[%d] : ", i);

scanf("%d", &a[i]);

}

}

void in(int a[], int n){

printf("Mang vua nhap : ");

for(int i = 0; i < n; i++){

printf("%d ", a[i]);

}

}

int main(){

//N là số lượng phần tử của mảng

int n;

printf("Nhap n : ");

int a[n]; //Khai báo mảng a vừa đủ n phần tử

scanf("%d", &n);

nhap(a, n);

in(a, n);

return 0;

}

Ví dụ 2 : Thay đổi giá trị trong mảng thông qua hàm.

#include <stdio.h>

void change(int a[], int n){

for(int i = 0; i < n; i++){

a[i] = 28;

}

}

int main(){

int m = 6;

int b[6] = {1, 2, 3, 4, 5, 6};

change(b, m);

for(int i = 0; i < m; i++){

printf("%d ", b[i]);

}

return 0;

}

Output :

28 28 28 28 28 28

Chú ý : Khi một hàm có tham số là mảng thì những thay đổi bạn thực hiện bên trong hàm đối với mảng tham số sẽ tương tự như với mảng đối số mà bạn truyền vào cho hàm.

**5. Chú Ý Khi Sử Dụng Mảng**

**Chú ý 1** :

Trong ngôn ngữ lập trình C thì chỉ số hợp lệ của mảng có N phần tử sẽ là từ 0 tới N - 1, tuy nhiên nó lại không cấm việc bạn truy cập vào các chỉ số không hợp lệ.

Ví dụ mảng có 100 phần tử nhưng bạn lại truy cập vào chỉ số 100, -1, hay 102...

Khi truy cập vào các chỉ số không hợp lệ trong mảng có thể gây lỗi bộ nhớ hoặc sai kết quả của bài toán vì các giá trị tại những chỉ số không hợp lệ này thường là giá trị rác và bạn không thể xác định giá trị của chúng.

Thói quen tốt khi sử dụng mảng đó là bạn cần quản lý code của mình không truy cập vào các chỉ số không hợp lệ.

**Chú ý 2 :**

Khai báo mảng kích thước tối đa là bao nhiêu, thông thường các bài toán liên quan tới dãy số sẽ cho các bạn dãy số không quá 106 phần tử.

Trong ngôn ngữ C bạn có thể khai báo mảng int cỡ khoảng 107 phần tử. Từ bộ nhớ tối đa cho phép của đề bài bạn có thể tính ra số lượng phần tử tối đa mà bạn có thể khai báo cho mảng, chứ không phải muốn khai báo bao nhiêu tùy ý :D

**Chú ý 3 :**

Khai báo mảng quá lớn trong hàm main sẽ bị tràn bộ nhớ stack

Ví dụ khi tràn bộ nhớ stack :

#include <stdio.h>

int main(){

printf("hello stack !\n");

int a[1000000];

return 0;

}

Output :

Process exited after 0.5098 seconds with return value 3221225725

Cách xử lý khi bạn muốn khai báo mảng cỡ 106 hoặc 107 phần tử mà không bị tràn stack đó là khai báo mảng ở phạm vi toàn cục

Khi bạn khai báo trong main thì vùng nhớ cấp phát để lưu mảng là vùng nhớ stack ( tương đối nhỏ) còn khi bạn khai báo mảng ngoài main thì vùng nhớ cấp phát cho mảng là vùng nhớ Uninitialized data segment. Vùng nhớ này có kích thước lớn hơn so với stack nên không bị tràn bộ nhớ khi mảng có kích thước lớn.

Code :

#include <stdio.h>

int a[1000000];

int main(){

printf("hello stack !\n");

return 0;

}

Output :

hello stack !