



APTECH SAIGON

HỆ THỐNG ĐÀO TẠO LẬP TRÌNH VIỆN QUỐC TẾ

Mảng

APTECH SAIGON



APTECH SAIGON

HỆ THỐNG ĐÀO TẠO LẬP TRÌNH VIÊN QUỐC TẾ

Cấu trúc mạng

• Khai báo mạng và sử dụng mạng

• Mạng 2 chiều, nhiều chiều

• Các tác vụ trên mạng

APTECH SAIGON

Định nghĩa mảng

Mảng: là tập hợp nhiều phần tử cùng kiểu có chung 1 tên.

Mỗi phần tử của mảng biểu diễn 1 giá trị.

Kiểu dữ liệu có thể là kiểu dữ liệu có sẵn hoặc kiểu dữ liệu do người dùng tự định nghĩa.

Các phần tử của Mảng được lưu trên những vùng nhớ liên tiếp nhau của bộ nhớ

Mỗi lần thực thi, địa chỉ đầu tiên của mảng được cấp ngẫu nhiên nhưng các phần tử tiếp theo là những vùng nhớ kế tiếp trong bộ nhớ

```
int Arr[4];  
for(int k=0; k<4; k++)  
    printf("\nDia chi phan tu %d la %d", k, &Arr[k]);
```

```
Dia chi phan tu 0 la 6487552  
Dia chi phan tu 1 la 6487556  
Dia chi phan tu 2 la 6487560  
Dia chi phan tu 3 la 6487564
```

Địa chỉ các phần tử liên tiếp cách nhau 4 byte (kích thước của kiểu int)

Các yếu tố để xác định mảng

Tên mảng

Kiểu mảng

Số chiều và kích thước mỗi chiều

Kích thước mảng = Tích của kích thước các chiều * sizeof(kiểu mảng)

Khai báo	Tên mảng	Kiểu dữ liệu	Số chiều – kích thước mỗi chiều
char a[5];	a	char	1 chiều, kích thước 5 phần tử
int b[3][4];	b	int	2 chiều, 1 chiều kích thước 3, 1 chiều kích thước 4
float x[3][10];	x	float	2 chiều, kích thước 3x10 phần tử
double diem[5][5][9]	diem	double	?

Các yếu tố để xác định mảng (tt)

int a[4]:

mảng kiểu int, tên a, số chiều 1, kích thước 4

Mảng có 4 phần tử là: a[0], a[1], a[2], a[3]

Mỗi phần tử a[i] của mảng chứa một giá trị kiểu int

Mảng a biểu diễn 1 dãy số nguyên

a[0]	a[1]	a[2]	a[3]
3	7	9	2

float b[2][3];

- Mảng kiểu float, tên là b, số chiều 2, kích thước các chiều là 2 và 3.
- Mảng có 6 phần tử được đánh số và sắp xếp:

	Cột 0	Cột 1	Cột 2
Dòng 0	b[0][0]	b[0][1]	b[0][2]
Dòng 1	b[1][0]	b[1][1]	b[1][2]

- Mỗi phần tử $b[i][j]$ chứa 1 giá trị kiểu float.
- Mảng b biểu diễn 1 bảng số thực 2 dòng 3 cột

Khai báo và sử dụng mảng

• Khai báo mảng

- Khai báo mảng 1 chiều: <kiểu dữ liệu> <tên biến mảng>[số phần tử];

<kiểu dữ liệu>: là kiểu dữ liệu của các phần tử trong mảng.

<số phần tử>: là số phần tử tối đa có thể có của mảng 1 chiều. Số phần tử phải là một số nguyên cụ thể hoặc là 1 hằng số nguyên.

Ví dụ: `int a[5]; char b[20];`

Khai báo mảng nhiều chiều:

<kiểu dữ liệu> <tên biến mảng>[<N1>][<N2>]...[<Nn>];

N1,N2,...Nn là số lượng phần tử tối đa của mỗi chiều (số nguyên cụ thể hoặc hằng số nguyên).

Ví dụ: `int matran[5][5]; float M[3][4][5];`

Truy cập phần tử mảng – khởi tạo giá trị mảng

Truy cập:

- Mỗi phần tử trong mảng đều có chỉ số dùng để truy cập. Chỉ số của mỗi phần tử là duy nhất.
- Phần tử đầu tiên luôn có chỉ số là 0, phần tử thứ 2 là 1,...
- Mỗi phần tử trong mảng có thể coi như một biến thuộc kiểu dữ liệu của mảng. Có thể thực hiện các thao tác nhận, xuất, gán,... lên phần tử mảng
- Sử dụng toán tử [] để truy xuất phần tử mảng. Cú pháp: tên_mảng[n1]; hoặc tên_mảng[n1][n2]: n1 và n2 là chỉ số của phần tử mảng (các số nguyên)

• Khởi tạo

- Mảng có thể được khởi tạo khi khai báo mảng

```
//khai báo hết tất cả các phần tử
int a[5] = {12,3,4,0,2};
//có thể khởi tạo chỉ một số phần tử
float b[3] = {1.5,2.7};
int x[10][3] = { { 1, 5, 7 },
                { 0 },
                { 2, 3 },
                { 1 }
              };
char KyTu[20] = {'H','a','p','p','y'};
```

- Hoặc khởi tạo bằng cách gán trực tiếp giá trị cho phần tử cụ thể

```
int a[4];
a[0] = 15; //phần tử thứ 0 =15
a[3] = 10; //phần tử thứ 3 =10
//các phần tử còn lại chưa khởi tạo
```

Khởi tạo giá trị tự động cho mảng

Sử dụng hàm rand() để tạo số ngẫu nhiên nguyên int: int x= rand()

Sử dụng rand() % (maxN + 1 - minN) + minN để tạo số ngẫu nhiên từ minN tới maxN

Sử dụng srand((int)time(0)); để thay đổi kết quả rand() sau mỗi lần thực thi

```
#include <stdio.h>
#define max 30
void main()
{
    int M1[max], n1=25;
    srand((int)time(0)); //time.h, stdlib.h
    // lấy số ngẫu nhiên từ -30 tới 30
    int minN=-30, maxN=30;
    for(int i=0; i<n1; i++)
        M1[i]= minN + rand() % (maxN + 1 - minN);
    printf("\nNoi dung mang M1:");
    for(int i=0; i<n1; i++)
        printf("%d ", M1[i]);
    return 0;
}
```

Sao chép mảng

Duyệt qua từng phần tử của mảng nguồn

Chép qua từng phần tử tương ứng của mảng đích

Thiết lập kích thước mảng đích bằng kích thước mảng nguồn

```
#include <stdio.h>
#define max 30
void main()
```

```
int M1[max], M2[max], n1, n2;
n1=5;
for(int i=0; i<n1; i++)
{
    printf("M1[%d]=", i);
    scanf("%d", &M1[i]);
}
for(int i=0; i<n1; i++) M2[i]= M1[i];
n2=n1;
printf("\nMang M2:");
for(int i=0; i<n2; i++) printf("%d ", M2[i]);
return 0;
```

```
//Ví dụ tạo mảng các số nguyên ngẫu nhiên từ 1 đến 30
#include <stdio.h>
#define max 30
void main()
{
```

```
int M1[max], M2[max], n1, n2;
n1=5;
srand((int)time(0));
for(int i=0; i<n1; i++) M1[i]= rand() %30;
for(int i=0; i<n1; i++)
    M2[i]= M1[i];
n2=n1;
printf("\nNoi dung mang M2:");
for(int i=0; i<n2; i++)
    printf("%d ", M2[i]);
return 0;
```

```
}
```

Mảng 2 chiều

Mảng hai chiều tương tự như một bảng tính

Chỉ số của mỗi phần tử chính là chỉ số dòng và chỉ số cột trong bảng tính.

Ví dụ: `int b[3][4];` // mảng 2 chiều tên b kiểu int. Chỉ số dòng = 3, chỉ số cột = 4

	<u>Cột 1</u>	<u>Cột 2</u>	<u>Cột 3</u>	<u>Cột 4</u>
<u>Dòng 1</u>	b[0][0]	b[0][1]	b[0][2]	b[0][3]
<u>Dòng 2</u>	b[1][0]	b[1][1]	b[1][2]	b[1][3]
<u>Dòng 3</u>	b[2][0]	b[2][1]	b[2][2]	b[2][3]

Truy xuất: `b[0][2] = 20; int x = b[1][0];`



Khởi tạo mảng 2 chiều

Khi khai báo: `int arr1[3][4] = {1,2,3,4,5,6,7,8,9,10,11,12};`
`int arr2[3][4] = {`
 `{1,2,3, 4},`
 `{5,6, 7, 8},`
 `{9, 10, 11, 12}`
`};`

Sử dụng vòng lặp:

```
int arr3[3][4]; for(int i=0; i<3; i++)
    for(int j=0; j<=4; j++) arr3[i][j] = i*j;
for (int i=0; i<3; i++)
{
    printf("\n");
    for(int j=0; j<4; j++) printf("%5d", arr1[i][j]);
}
for (int i=0; i<3; i++)
{
    printf("\n");
    for(int j=0; j<4; j++) printf("%05d ", arr2[i][j]);
}
```



Khởi tạo mảng 2 chiều

•/Khởi tạo ngẫu nhiên mảng 2 chiều (ma trận) các số nguyên

```
#include<stdio.h>
#define max 10
int main(){
    srand( (int)time(0) );
    int M[max][max], n=5, i, j, minN=-5, maxN=20;
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            M[i][j]= rand() %(maxN+1-minN) + minN;
    printf("\n Ma tran \n");
    for(i=0; i<n; i++)
    {
        printf("\n");
        for(j=0; j<n; j++)
            printf("%4d", M[i][j]);
    }
}
```



Khởi tạo mảng 2 chiều

// Khởi tạo ngẫu nhiên mảng 2 chiều các số thực

```
#include<stdio.h>
#include <stdlib.h>
#include<time.h>
#define max 10
int main(){
    srand( (int)time(0) );
    float M[max][max]; int n=5, i, j, minN=-20, maxN=100;
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            M[i][j] = (rand() / (float) RAND_MAX) *(maxN - minN) + minN;
    printf("\n Ma tran \n");
    for(i=0; i<n; i++)
    {
        printf("\n");
        for(j=0; j<n; j++)
            printf("%10.2f", M[i][j]);
    }
}
```



Các thao tác trên mảng

Nhập xuất mảng

Kiểm tra tính chất của mảng

Tìm kiếm trên mảng

Đếm phần tử

Tính toán có điều kiện

Thêm, sửa, xóa, tách ghép mảng

sắp xếp mảng

APTECH SAIGON

Bài tập mảng

Viết chương trình nhập 1 mảng các số nguyên và

- Xuất mảng theo thứ tự nhập, ngược thứ tự nhập
- Tìm phần tử lớn nhất, nhỏ nhất,
- Tìm vị trí phần tử âm, dương đầu tiên
- Kiểm tra mảng có phải mảng đối xứng hay không
- Đếm số phần tử âm, dương, chẵn, lẻ, số nguyên tố, số chính phương,...
- Tính tổng các phần tử dương, tích các phần tử có giá trị từ 5 đến 10
- Tách mảng ra thành 2 mảng: mảng chứa số phần tử chẵn, mảng chứa số phần tử lẻ
- Đảo ngược mảng
- Xóa phần tử đầu, phần tử cuối, phần tử có chỉ số k bất kỳ ($0 \leq k < n$)
- Sắp xếp mảng: tang dần, giảm dần



String

String: Một chuỗi có thể được định nghĩa là một mảng kiểu ký tự, được kết thúc bởi một ký tự null.

Mỗi ký tự trong chuỗi chiếm một byte (char) và ký tự cuối cùng của một chuỗi là “\0”

```
#include <stdio.h>
void main()
{
    char ary[5];
    int i;
    printf("\n Enter string : ");
    scanf("%s", ary);
    printf("\n The string is %s \n\n", ary);
    for (i=0; i<5; i++)
        printf("\t%d", ary[i]);
}
```



Các hàm trên String

strlen(s)	Trả về chiều dài chuỗi s1
strcmp(s1, s2)	Trả về 0 nếu s1 và s2 giống nhau. <0 nếu s1<s2, >0 nếu s1>s2
strstr(s1, s2)	Trả về con trỏ tới vị trí đầu tiên của s2 trong s1
strchr(s1, ch)	Trả về con trỏ tới vị trí đầu tiên của ch trong s1
strcpy(s1, s2)	Sao chép s2 vào s1
strcat(s1, s2)	Nối s2 vào cuối s1