



Mảng

Phiên 7

Chỉ dành cho Trung tâm Aptech sử dụng



Mục tiêu

- Giải thích các phần tử và chỉ số của mảng
- Định nghĩa một mảng
- Giải thích cách xử lý mảng trong C
- Giải thích cách một mảng được khởi tạo
- Giải thích về mảng chuỗi/ký tự
- Giải thích mảng hai chiều
- Giải thích khởi tạo mảng hai chiều



Các phần tử và chỉ số của mảng

- Mỗi thành viên của một mảng được xác định bằng chỉ số hoặc chỉ số dưới duy nhất được gán cho nó
- Kích thước của một mảng được xác định bởi số lượng chỉ số cần thiết để xác định duy nhất từng phần tử
- Chỉ số là một số nguyên dương được bao trong [] và được đặt ngay sau tên mảng
 - Một chỉ mục chứa các giá trị số nguyên bắt đầu bằng số không.
 - Một mảng có 11 phần tử sẽ trông như thế này -

Người chơi[0], người chơi[1], người chơi[2],.... Người chơi[10]



Định nghĩa một Mảng-1

- Một mảng có một số đặc điểm riêng biệt và phải được định nghĩa với chúng
- Những đặc điểm này bao gồm –

Lớp lưu trữ

Kiểu dữ liệu của các phần tử trong Mảng

Tên Mảng

Chỉ ra vị trí của thành viên đầu tiên của mảng

Kích thước mảng

một hằng số đánh giá thành giá trị +ve



Định nghĩa một Mảng-2

Mảng được định nghĩa theo cùng cách như biến được định nghĩa. Sự thay đổi duy nhất là tên mảng được theo sau bởi một hoặc nhiều biểu thức, được đặt trong dấu ngoặc vuông [], chỉ định chiều mảng.

Kiểu dữ liệu `Storage_Class array_name[size]`

`int người chơi[11];`



Chuẩn mực với Mảng

- Tất cả các phần tử của một mảng đều có cùng kiểu
- Mỗi phần tử của một mảng có thể được sử dụng ở bất cứ nơi nào một biến được phép hoặc yêu cầu
- Mỗi phần tử của một mảng có thể được tham chiếu bằng cách sử dụng một biến hoặc một biểu thức số nguyên
- Mảng có thể có các kiểu dữ liệu như sau **int, char, float hoặc double**



Xử lý mảng trong C-1

- Mảng được xử lý khác với biến trong C
- Hai mảng, ngay cả khi chúng có cùng kiểu và kích thước, cũng không thể được kiểm tra xem có bằng nhau hay không
- Không thể gán một mảng trực tiếp cho một mảng khác
- Giá trị không thể được gán cho toàn bộ một mảng, thay vào đó, giá trị được gán cho các phần tử của mảng



Xử lý mảng trong C-2

/* Các giá trị đầu vào được chấp nhận từ người dùng vào mảng ary[10]*/

bao gồm <stdio.h>

hàm main() không có giá trị

```
{  
    int ary[10];  
    int i, tổng, cao; for(i=0;  
        i<10; i++)  
    {  
        printf("\n Nhập giá trị: %d : ", i+1);  
        scanf("%d",&ary[i]);  
    }  
    /* Hiển thị giá trị cao nhất được nhập */ high = ary[0];  
  
    đối với (i = 1;  tôi <10;  tôi++)  
    {  
        nếu(ary[i]  > cao)  
            cao =  ary[i];  
    }  
    printf("\n Giá trị cao nhất được nhập là %d", cao); /* in giá trị  
    trung bình của các giá trị được nhập cho ary[10] */  
    for(i=0, total=0; i<10; i++)  
        tổng = tổng + ary[i];  
    printf("\n Giá trị trung bình của các phần tử của ary là %d", total/i);  
}
```




Khởi tạo mảng

- Mỗi phần tử của mảng Automatic cần được khởi tạo riêng biệt
- Trong ví dụ sau, các phần tử mảng đã được gán giá trị bằng cách sử dụng **vòng lặp**
bao gồm <stdio.h>

hàm main() không có giá trị

```
{  
    ký tự alpha[26];  
    số nguyên i, j;  
    đối với (i = 65, j = 0; i < 91; i++, j++) {  
  
        alpha[j] = i;  
        printf("Ký tự được gán bây giờ là %c \n", alpha[j]); }  
  
    lấy char();  
}
```

- Trong trường hợp mảng extern và static, các phần tử được tự động khởi tạo thành 0

Chuỗi/Ký tự Mảng-1

- Một chuỗi có thể được định nghĩa là một mảng kiểu ký tự, được kết thúc bằng một ký tự null
- Mỗi ký tự trong chuỗi chiếm một byte và ký tự cuối cùng của chuỗi là “\0” (Dấu gạch chéo ngược số không)
- Ví dụ

bao gồm <stdio.h>

hàm main() không có giá trị

{

char ary[5];

số nguyên i;

printf("\n Nhập chuỗi: ");

scanf("%s", ary);

printf("\n Chuỗi là %s \n\n", ary); for (i=0; i<5; i++)

printf("\t%d", ary[i]);

}

Chuỗi/Ký tự Mảng-2

Đầu ra -

If the entered string is appl, the output will be as shown below.

```
The string is appl
97  112 112 108  0
```

**Đầu vào cho phần trên là 4 ký tự và 5th
ký tự là ký tự null**

If the entered string is apple, the output will be as shown below.

```
The string is apple
97  112 112 108 101
```

Đầu ra ở trên dành cho đầu vào 5 ký tự



Các hàm chuỗi

Nhiều hàm chuỗi có trong
tập tin tiêu đề chuẩn `<string.h>`

Name	Function
<code>strcpy(s1, s2)</code>	Copies <code>s2</code> into <code>s1</code>
<code>strcat(s1, s2)</code>	Concatenates <code>s2</code> onto the end of <code>s1</code>
<code>strlen(s1)</code>	Returns the length of <code>s1</code>
<code>strcmp(s1, s2)</code>	Returns 0 if <code>s1</code> and <code>s2</code> are the same; less than 0 if <code>s1 < s2</code> ; greater than 0 if <code>s1 > s2</code>
<code>strchr(s1, ch)</code>	Returns a pointer to the first occurrence of <code>ch</code> in <code>s1</code>
<code>strstr(s1, s2)</code>	Returns a pointer to the first occurrence of <code>s2</code> in <code>s1</code>



Mảng hai chiều

- Mảng đa chiều đơn giản nhất và được sử dụng phổ biến nhất là mảng hai chiều - mảng chiều
- Một mảng hai chiều có thể được coi như một mảng gồm hai mảng một chiều
- Mảng hai chiều trông giống như bảng thời gian biểu của tàu hỏa bao gồm các hàng và cột
- Mảng hai chiều được khai báo là -

```
int temp[4][3];
```



Khởi tạo của Mảng đa chiều-1

```
int ary[3][4] =
```

```
{1,2,3,4,5,6,7,8,9,10,11,12};
```

Kết quả của bài tập trên sẽ như sau:

```
ary [0] [0] = 1
```

```
ary [0] [1] = 2
```

```
ary [0] [2] = 3
```

```
ary [0] [3] = 4
```

```
ary [1] [0] = 5
```

```
ary [1] [1] = 6
```

```
ary [1] [2] = 7
```

```
ary [1] [3] = 8
```

```
ary [2] [0] = 9
```

```
ary [2] [1] = 10
```

```
ary [2] [2] = 11
```

```
ary [2] [3] = 12
```



Khởi tạo của Mảng đa chiều-2

```
int ary[3][4]=  
    {  
        {1,2,3},  
        {4,5,6},  
        {7,8,3}  
    };
```



Khởi tạo của Mảng đa chiều-3

Kết quả của bài tập sẽ như sau:

<code>ary[0][0] =1</code>	<code>ary[0][1]=2</code>	<code>ary[0][2]=3</code>	<code>ary[0][3]=0</code>
<code>ary[1][0]=4</code>	<code>ary[1][1]=5</code>	<code>ary[1][2]=6</code>	<code>ary[1][3]=0</code>
<code>ary[2][0]=7</code>	<code>ary[2][1]=8</code>	<code>ary[2][2]=3</code>	<code>ary[2][3]=0</code>

Mảng chuỗi hai chiều được khai báo theo
cách sau:

```
char str_ary[25][80];
```




Mảng hai chiều-1

```
# bao gồm <stdio.h>
# bao gồm <string.h>
void main ()
{
    số nguyên i, n = 0;
    mục int;
    ký tự x[10][12];
    ký tự temp[12];

    clrscr();
    printf("Nhập mỗi chuỗi trên một dòng riêng biệt\n\n");
    printf("Nhập 'END' khi kết thúc \n\n");

    /* đọc danh sách các chuỗi */ thực hiện

    {
        printf("Chuỗi: %d : ", n+1);
        scanf("%s" x[n]);
    } while (strcmp(x[n++], "END")); /* sắp
xếp lại danh sách các chuỗi */
```

Ví dụ

tiếp theo....

Mảng hai chiều-2

```
n = n - 1;  
    đối với (mục = 0; mục < n-1; ++ mục) {
```

```
        /* tìm chuỗi thấp nhất còn lại */  
        đối với (i = mục + 1; i < n; ++ i) {
```

```
            nếu(strcmp(x[item], x[i]) > 0) {
```

```
                /*hoán đổi hai sting */ strcpy  
                (temp, x[item]); strcpy (x[item],  
                x[i]); strcpy (x[i], temp);
```

```
            }
```

```
        }
```

```
    }
```

```
    /* Hiển thị danh sách các chuỗi đã sắp xếp */  
    printf("Danh sách các chuỗi đã ghi lại: \n");  
    đối với (i = 0; i < n; ++i) {
```

```
        printf("\nChuỗi %d là %s", i+1, x[i]);
```

```
    }
```

```
}
```

Ví dụ