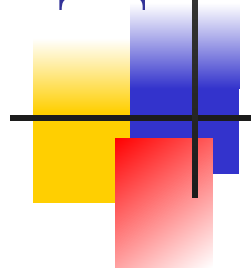




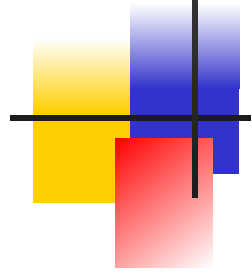
Mảng

Bài 7



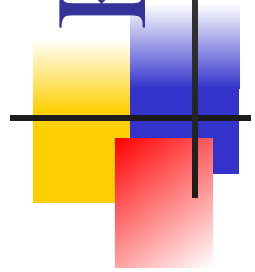
Mục tiêu của bài học

- Các phần tử của mảng và các chỉ số
- Khai báo mảng
- Cách quản lý mảng trong C
- Cách khởi tạo mảng
- Tìm hiểu chuỗi / mảng ký tự
- Tìm hiểu mảng hai chiều
- Cách khởi tạo mảng hai chiều



Các phần tử và chỉ số của mảng

- Mỗi phần tử được xác định bằng một số thứ tự (còn gọi là chỉ số) duy nhất trong mảng
- Số chiều của mảng được xác định bằng số các chỉ số cần thiết để định danh duy nhất từng phần tử
- Chỉ số là một số nguyên dương trong [] đặt ngay sau tên mảng
- Chỉ số của mảng (trong C) được bắt đầu là 0
- Mảng *player* với 11 phần tử :
player[0], player[1], player[2],..., player[10]



Khai báo mảng

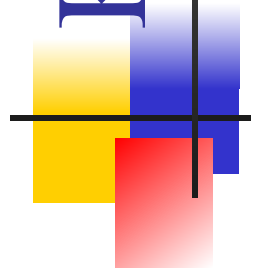
- Các đặc tính riêng của mảng cần được định nghĩa.

Lớp lưu trữ

Kiểu dữ liệu của các phần tử

Tên mảng đại diện cho vị trí phần tử đầu tiên

Kích thước mảng một hằng số



Khai báo mảng (tt.)

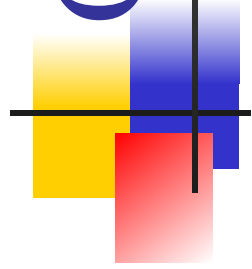
Khai báo mảng giống như cách khai báo biến. Chỉ khác là tên mảng được theo sau bởi một hoặc nhiều biểu thức đặt trong cặp dấu ngoặc vuông [], để xác định kích thước của mảng.

```
int player[11];
```



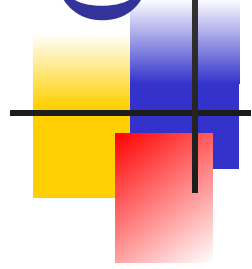
Các qui tắc

- Các phần tử của mảng có cùng kiểu dữ liệu
- Mỗi phần tử của mảng có thể được sử dụng như một biến riêng lẻ
- Kiểu dữ liệu của mảng có thể là **int, char, float** hoặc **double**



Quản lý mảng trong C

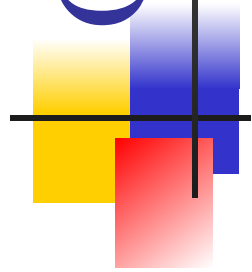
- Trong ngôn ngữ C, mảng được “đổi xử” không giống hoàn toàn với biến
- Hai mảng có cùng kiểu và cùng kích thước cũng không được xem là tương đương nhau
- Không thể gán trực tiếp một mảng cho một mảng khác.
- Không thể gán trị cho toàn bộ mảng, mà phải gán trị cho từng phần tử của mảng



Quản lý mảng trong C (tt)

/*Input values are accepted from the user
into the array ary[10]*/

```
#include <stdio.h>
void main() {
    int ary[10];
    int i, total, high;
    for(i=0; i<10; i++) {
        printf("\n Enter value: %d : ",
            i+1);
        scanf("%d", &ary[i]);
    }
```

Quản lý mảng trong C (tt)

```
/* Displays highest of the entered values */
high = ary[0];
for(i=1; i<10; i++){
    if(ary[i] > high) high = ary[i];
}
printf("\nHighest value entered was %d", high);

/* prints average of values entered for ary[10] */
for(i=0, total=0; i<10; i++) total = total +
ary[i];
printf("\nThe average of the elements of ary
is%d", total/i);
}
```



Khởi tạo mảng

- Mỗi phần tử của một mảng auto cần được khởi tạo riêng rẽ.
- Trong ví dụ sau các phần tử của mảng được gán giá trị bằng cách sử dụng vòng lặp **for**

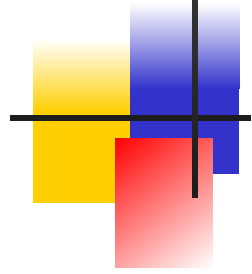
```
#include <stdio.h>

void main() {
    char alpha[26];
    int i, j;
    for(i=65, j=0; i<91; i++, j++) {
        alpha[j] = i;
        printf("The character now assigned
        is%c\n", alpha[j]);
    }
    getchar();
}
```



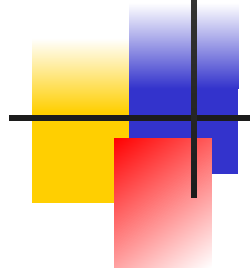
Khởi tạo mảng (tt)

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



Chuỗi/Mảng ký tự

- Chuỗi có thể được định nghĩa như là một mảng kiểu ký tự, được kết thúc bằng ký tự null
- Mỗi ký tự trong chuỗi chiếm một byte và ký tự cuối cùng của chuỗi là “\0” (null)
- Ví dụ:



Chuỗi/Mảng ký tự (ví dụ)

```
#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]);
}
```



Chuỗi/Mảng ký tự (tt)

Chạy chương trình:

Enter string:

Nếu dữ liệu nhập là “appl”, output của chương trình là:

The string is appl

97 112 112 108 0



Các hàm xử lý chuỗi

- Các hàm xử lý chuỗi được tìm thấy trong thư viện chuẩn <string.h>

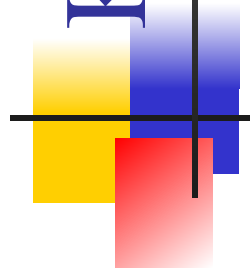
Name	Function
strcpy(s1, s2)	Copies s2 into s1
strcat(s1, s2)	Concatenates s2 onto the end of s1
strlen(s1)	Returns the length of s1
strcmp(s1, s2)	Returns 0 if s1 and s2 are the same; less than 0 if s1 < s2; greater than 0 if s1 > s2
strchr(s1, ch)	Returns a pointer to the first occurrence of ch in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



Mảng hai chiều

- Mảng đa chiều đơn giản nhất và thường được dùng nhất là mảng hai chiều
- Mảng hai chiều có thể xem như là một mảng với mỗi phần tử là mảng một chiều
- Về logic, một mảng hai chiều trông giống như một bảng lịch trình xe lửa, gồm các dòng và các cột
- Khai báo mảng hai chiều:

int temp[4][3];

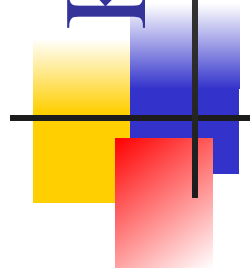


Khởi tạo mảng đa chiều

```
int ary[3][4] =  
    {1,2,3,4,5,6,7,8,9,10,11,12};
```

Kết quả của phép gán trên 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

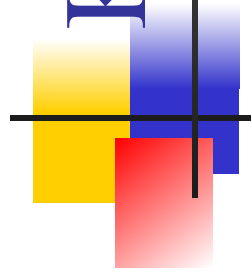


Khởi tạo mảng đa chiều (tt)

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

Kết quả của phép gán trên như sau:

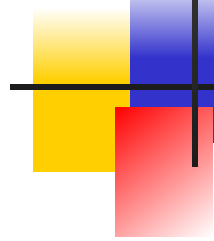
ary[0][0] = 1	ary[0][1] = 2	ary[0][2] = 3	ary[0][3] = 0
ary[1][0] = 4	ary[1][1] = 5	ary[1][2] = 6	ary[1][3] = 0
ary[2][0] = 7	ary[2][1] = 8	ary[2][2] = 3	ary[2][3] = 0



Khởi tạo mảng đa chiều (tt)

Một mảng chuỗi hai chiều được khai báo theo cách sau:

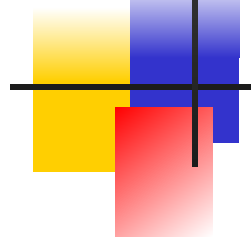
```
char str_ary[25][80];
```



Mảng hai chiều - Ví dụ

```
#include <stdio.h>
#include <string.h>
void main () {
    int i, n = 0;
    int item;
    char x[10][12];
    char temp[12];
    clrscr();
    printf("Enter each string on a separate
line\n\n");
    printf("Type 'END' when over \n\n");
    /* read in the list of strings */
    do {
        printf("String %d : ", n+1);
        scanf("%s", x[n]);
    } while (strcmp(x[n++], "END"));
    /*reorder the list of strings */
```

còn tiếp...



Mảng hai chiều - Ví dụ (tt.)

```
n = n - 1;
for(item=0; item<n-1; ++item) {
    /* find lowest of remaining strings */
    for(i=item+1; i<n; ++i) {
        if(strcmp (x[item], x[i]) > 0){
            /*interchange two stings */
            strcpy (temp, x[item]);
            strcpy (x[item], x[i]);
            strcpy (x[i], temp);
        }
    }
}
/* Display the arranged list of strings */
printf("Recorded list of strings : \n");
for(i = 0; i < n ; ++i) {
    printf("\nString %d is %s", i+1, x[i]);
}
}
```