

KỸ THUẬT LẬP TRÌNH C

Chương 6: Mảng một chiều

1. Khái niệm mảng
2. Khai báo mảng một chiều
3. Khởi tạo mảng một chiều
4. Nhập xuất mảng một chiều
5. Dùng mảng một chiều làm tham số truyền cho hàm

1. Khái niệm mảng

- Mảng là một tập hợp nhiều phần tử (biến) có cùng một kiểu và chung một tên được sắp xếp liên tiếp nhau trong bộ nhớ.
- Mỗi phần tử của mảng có một đại lượng xác định vị trí tương đối của phần tử đó so với các phần tử khác trong mảng, gọi là **chỉ số**.

2. Khai báo mảng một chiều

Cú pháp:

<Kiểu> <Tên mảng>[<Kích thước>];

- Mỗi phần tử của mảng được truy nhập thông qua tên mảng cùng với chỉ số đặt giữa hai ngoặc vuông.
- Chỉ số là một số nguyên được đánh số từ 0 đến <Kích thước> - 1.

Ví dụ:

```
int a[6];
```

Câu lệnh trên khai báo một mảng a gồm 6 phần tử kiểu int, bao gồm:

a[0], a[1], a[2], a[3], a[4], a[5]

3. Khởi tạo mảng một chiều

- Để khởi tạo mảng ta liệt kê danh sách các giá trị của chúng trong cặp dấu ngoặc “{” và “}”.
- Kích thước mảng, nếu có, không được nhỏ hơn số giá trị có trong danh sách khởi tạo.

Ví dụ:

```
int a[5] = {20, -5, 8, 40, 10};
```

```
int b[] = {100, -25, 18, 10, 18};
```

Ví dụ 1: Nhập vào một số nguyên dương n ($1 \leq n \leq 7$). Tùy theo n , hãy in ra các từ Sun, Mon, ..., Sat tương ứng.

```
void main() {  
    char *a[7]= {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri",  
                 "Sat"}; //Khởi tạo mảng các chuỗi  
  
    int n;  
    <Nhập n>  
    if(n >= 1 && n <= 7)  
        printf("%s\n", a[n - 1]);  
    else printf("DL nhap khong hop le\n");  
}
```

Ví dụ 2: Nhập tháng và năm. Hãy tính và in ra số ngày trong tháng.

```
void main() {  
    int m, y;  
    <Nhập m, y>  
    int a[12] = { 31, Nhuận(y)? 29 : 28, 31, 30, 31, 30,  
                 31, 31, 30, 31, 30, 31}; /*Khởi tạo mảng các  
                                         số nguyên*/  
    if(m > 0 && m < 13 && y > 0)  
        printf("Số ngày trong tháng %d năm %d là  
              %d\n", m, y, a[m - 1]);  
    else  
        printf("DL nhập không hợp lệ\n");  
}
```


4. Nhập xuất mảng một chiều

Ví dụ:

```
#define SIZE 20
```

```
void main() {
```

```
    int a[SIZE]; /*khai báo mảng gồm SIZE phần tử  
                  kiểu int*/
```

```
    int n; // Lưu số phần tử thực có của mảng
```

```
    int i;
```

```
do {  
    printf("Nhap so phan tu:");  
    scanf("%d", &n);  
} while(n < 1 || n > SIZE);
```

```
// Nhập dữ liệu cho mảng  
for(i = 0; i < n; i++) {  
    printf("pt thu %d:", i);  
    scanf("%d", &a[i]);  
}
```

```
// Xuất dữ liệu của mảng  
for(i = 0; i < n; i++)  
    printf("%d\t", a[i]);  
printf("\n");  
}
```

5. Dùng mảng một chiều làm tham số truyền cho hàm

- Tên mảng là một **hằng địa chỉ** và nó chính là **địa chỉ phần tử đầu tiên** của mảng.
- Khi dùng tên mảng làm tham số thực truyền cho hàm thì thực chất là địa chỉ phần tử đầu tiên của mảng được truyền cho hàm và như vậy tham số hình thức tương ứng trong định nghĩa hàm phải viết dưới dạng con trỏ.

Ví dụ:

```
#define SIZE 20
// khai báo các nguyên mẫu hàm
void Nhap(int *a, int *n);
void Xuat(int *a, int n);
// Định nghĩa các hàm
void Nhap(int *a, int *n) {
    do {
        printf("Nhap so phan tu:");
        scanf("%d", &(*n));
    } while(*n < 1 || *n > SIZE);
    for(int i = 0; i < *n; i++) {
        printf("pt thu %d:", i);
        scanf("%d", &a[i]);
    }
}
```

```
void Xuat(int *a, int n) {  
    for(int i = 0; i < n; i++)  
        printf("%d\t", a[i]);  
    printf("\n");  
}  
void main() {  
    int a[SIZE], n;  
    Nhap(a, &n);  
    Xuat(a, n);  
}
```

Lưu ý: Tham số hình thức tương ứng với tham số thực là tên mảng cũng có thể viết như sau:

```
void Nhap(int a[], int *n);
```

```
void Xuat(int a[], int n);
```

Hết