

Mục tiêu

Ôn tập lại kiến thức về mảng đã học trong môn Nhập môn lập trình.

Nội dung

Kiểu dữ liệu mảng trong C/C++

- Lưu ý về khai báo

```
int mang1[];  
int mang2[100];
```

Khai báo **mang1** là sai lỗi cú pháp vì thiếu giá trị số lượng phần tử tối đa.

Khai báo **mang2** là đúng.

Với C++ có thể khai báo như sau:

```
int mang1[100], mang2, mang3[100];
```

Lúc này thì phải lưu ý rằng biến **mang2** *chỉ là 1 biến int chứ không phải là biến mảng*.

- Lưu ý về cách truy suất mảng

```
int mang1[10];  
mang1[10] = 100;  
printf("mang1[10]= %d", mang1[10]);
```

Với đoạn chương trình như trên thì giá trị xuất ra sẽ đúng (về hiển thị). Tuy nhiên chương trình sẽ bị crash vì đã truy suất mảng không hợp lý. Lý do là vì số lượng phần tử tối đa của **mang1** là **10** (chỉ số truy suất mảng bắt đầu từ 0) nên truy suất **mang1[10]** là truy suất ngoài mảng.

- Về bản chất biến mảng là lưu giữ địa chỉ phần tử đầu của mảng (dãy các phần tử liên tiếp nhau). Do đó có thể truy suất mảng như con trỏ.

```
int mang1[10];  
*(mang1 + 3) = 100;  
printf("mang1[3]= %d", mang1[3]);
```

Phát sinh số ngẫu nhiên trong C++

- Tìm hiểu qua ví dụ sau và lưu ý các phần được tô.

```
#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
#include <conio.h>
```

```
void main()
{
    srand((unsigned int)time(NULL));
    int n = 3 + rand() % 10;
    for (int i=0; i<n; ++i)
    {
        printf("%d\n", rand());
    }
    getch();
}
```

Ví dụ hình thức đề thi Thực hành

Yêu cầu đề bài

Hoàn thành cài đặt các hàm cần thiết để hàm main chạy đúng (không được sửa đổi hàm main).

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAXLEN 100

void XuatMang(int mang[], int n)
{
    // code ở đây
}

void DaoMang(int mang[], int n)
{
    // code ở đây
}

void DtoB(long d, int mangKySo[], int &soKySo)
{
    // code ở đây
}

long BtoD(int mangKySo[], int soKySo)
{
    // code ở đây, nhớ thay đổi dòng return
    return 0;
}

void main()
{
    srand((unsigned int)time(NULL));

    long d = 163 + rand() % 10000;
    int mang[MAXLEN];
    int n = 0;

    // chuyển đổi sang nhị phân
    DtoB(d, mang, n);
    printf("Chuyen doi d= %d sang nhi phan: ", d);
}
```

```
XuatMang(mang, n);
printf("\n");

// chuyển đổi sang thập phân
DaoMang(mang, n);
long dKq = BtoD(mang, n);
printf("Chuyen doi b= ");
XuatMang(mang, n);
printf(" sang thap phan: %d\n", dKq);
}
```

Hướng dẫn

Chỉ cài đặt ở các hàm (có thể thêm code ngoài, hàm ngoài nhưng không được thay đổi hàm main)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define MAXLEN 100

void XuatMang(int mang[], int n)
{
    // code ở đây
    for (int i = 0; i < soKySo; i++)
    {
        printf("%d", mangKySo[i]);
    }
}

void DaoMang(int mang[], int n)
{
    // code ở đây
    int temp = 0;
    for (int i = 0; i < n / 2; i++)
    {
        temp = mang[i];
        mang[i] = mang[n - i - 1];
        mang[n - i - 1] = temp;
    }
}

void DtoB(long d, int mangKySo[], int &soKySo)
{
    // code ở đây
    int n = 0;
    // mảng phụ lấy ký số
    int mangPhu[MAXLEN];
    while (d > 0)
    {
        // lấy ký số
        mangPhu[n] = d % 2;
        // tăng biến đếm
        n++;
        // dịch chuyển
    }
}
```

```
        d /= 2;
    }
    // đổ lại vào mảng kết quả
    for (int i = 0; i < n; i++)
    {
        mangKySo[i] = mangPhu[n - i - 1];
    }
    soKySo = n;
}

long BtoD(int mangKySo[], int soKySo)
{
    // code ở đây, nhớ thay đổi dòng return
    long d = 0;
    int tempP = 1;
    for (int i = soKySo - 1; i >= 0; i--)
    {
        d += mangKySo[i] * tempP;
        tempP *= 2;
    }
    return d;
}

void main()
{
    srand((unsigned int)time(NULL));

    long d = 163 + rand() % 10000;
    int mang[MAXLEN];
    int n = 0;

    // chuyển đổi sang nhị phân
    DtoB(d, mang, n);
    printf("Chuyen doi d= %d sang nhi phan: ", d);
    XuatMang(mang, n);
    printf("\n");

    // chuyển đổi sang thập phân
    DaoMang(mang, n);
    long dKq = BtoD(mang, n);
    printf("Chuyen doi b= ");
    XuatMang(mang, n);
    printf(" sang thap phan: %d\n", dKq);
}
```

Bài tập

Bài 1 (mảng 1 chiều)

Thực hiện các yêu cầu sau:

- Nhập, xuất một mảng các số nguyên với số lượng phần tử cho trước.

Thực hành Kỹ thuật lập trình

- Viết hàm phát sinh mảng (số lượng phần tử ít nhất là 3, giá trị mỗi phần tử nằm trong khoảng cho trước).
- Tìm phần tử lớn nhất và nhỏ nhất của mảng.
- Tính tổng tất cả các phần tử của mảng.
- Tính tổng tất cả các phần tử có số lần xuất hiện lớn nhất.
- Tìm phần tử trung vị của mảng.
- Tìm mảng con của mảng (xác định 1 mảng có là con của mảng đã có hay không).

Bài 2 (mảng 1 chiều)

Viết chương trình chuyển đổi cơ số của số nguyên:

- Thập phân sang nhị phân và ngược lại.
- Thập phân sang Thập lục phân và ngược lại.

Bài 3 (mảng 2 chiều)

Viết chương trình phát sinh ma trận với số hàng và số cột cho trước (các giá trị nằm trong khoảng cho trước). Sau đó thực hiện các yêu cầu sau:

- Viết hàm đảo ma trận (thay thế giữa dòng và cột của ma trận).
- Viết hàm tìm kiếm trên ma trận.
- Tìm ma trận con trong ma trận đã có (xác định 1 ma trận có là con của ma trận đã có hay không).
- Viết hàm tính tổng các phần tử biên của ma trận.
- Viết hàm dịch xoay vòng cột của ma trận.