

Bài 34: CÁC THAO TÁC TRÊN MẢNG MỘT CHIỀU

Xem bài học trên website để ủng hộ Kteam: [Các thao tác trên Mảng một chiều](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài học trước, mình đã chia sẻ cho các bạn về khái niệm và cách sử dụng [MẢNG MỘT CHIỀU TRONG C++ \(Arrays\)](#).

Hôm nay, mình sẽ giới thiệu cho các bạn về **Các thao tác trên mảng một chiều trong C++**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về:

- [VÒNG LẶP FOR TRONG C++ \(For statements\)](#)
- [MẢNG MỘT CHIỀU TRONG C++ \(Arrays\)](#)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Truyền mảng vào hàm (passing arrays to functions)
- Nhập và xuất mảng 1 chiều
- Sao chép mảng 1 chiều
- Tìm kiếm phần tử trong mảng
- Sắp xếp mảng 1 chiều
- Thêm và xóa một phần tử trong mảng

Truyền mảng vào hàm (passing arrays to functions)

Trong bài [TRUYỀN GIÁ TRỊ CHO HÀM \(Passing Arguments by Value\)](#), bạn đã biết khi một biến truyền vào hàm theo phương pháp **truyền giá trị**, C++ sẽ **sao chép giá trị của đối số vào tham số** của hàm. Vì tham số hàm là 1 bản sao, nên việc **thay đổi giá trị tham số không làm thay đổi giá trị đối số ban đầu**.

Đối với kiểu dữ liệu mảng, việc sao chép 1 số lượng lớn các phần tử sẽ gây **tốn rất nhiều vùng nhớ** và **giảm hiệu suất**. Nên khi truyền mảng vào hàm, **tham số của hàm** chính là **địa chỉ vùng nhớ của phần tử đầu tiên trong mảng**. Vì vậy, mảng **có thể thay đổi nội dung** sau khi thực hiện hàm

Lưu ý khi truyền mảng vào hàm:

- Tham số kiểu mảng trong khai báo hàm **giống như khai báo biến mảng**.

```
void NhapMang(int a[100]);
```

- Tham số kiểu mảng truyền cho hàm chính là **địa chỉ của phần tử đầu tiên** của mảng.
 - Có thể **bỏ số lượng phần tử** hoặc **sử dụng con trỏ**.
 - Mảng **có thể thay đổi nội dung** sau khi thực hiện hàm.

```
void NhapMang(int a[]);  
void NhapMang(int *a);
```

- Số lượng phần tử thực sự truyền qua biến khác.

```
void NhapMang(int a[100], int n);  
void NhapMang(int a[], int n);  
void NhapMang(int *a, int n);
```

Tất cả các ví dụ bên dưới sẽ sử dụng phương pháp truyền mảng vào hàm.

Nhập và xuất mảng 1 chiều

Bên dưới là ví dụ về nhập, xuất dữ liệu cho mảng 1 chiều:

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
using namespace std;

// định nghĩa số phần tử mảng
#define MAX 1000

// khai báo prototype
void nhapMang(int arr[], int &n);
void xuatMang(int arr[], int n);

int main()
{
    int myArray[MAX]; // mảng myArray có MAX phần tử
    int nSize; // nSize là số phần tử được sử dụng, do user nhập

    // nhập xuất mảng tự động
    nhapMang(myArray, nSize);
    xuatMang(myArray, nSize);

    return 0;
}

// hàm nhập mảng
void nhapMang(int arr[], int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

    cout << "Nhập số lượng phần tử n : ";
    cin >> n;

    // khởi tạo ngẫu nhiên từng phần tử từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand();
    }
}
```

```
}

// hàm xuất mảng
void xuatMang(int arr[], int n)
{
    // xuất từng phần tử cho mảng từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        cout << "arr[" << i << "] = " << arr[i] << endl;
    }
}
```

Output:



Sao chép mảng 1 chiều

Ý tưởng: Để tạo ra một bản sao từ một mảng, bạn cần khai báo thêm 1 mảng khác có cùng kích thước với mảng ban đầu.

Chương trình:

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
#include <string>
using namespace std;

// định nghĩa số phần tử mảng
#define MAX 1000
```

```
// khai báo prototype
void nhapMang(int arr[], int &n);
void xuatMang(int arr[], int n);
void saoChepMangMotChieu(int arrDest[], int arrSource[], int n);

int main()
{
    int myArray[MAX]; // mảng myArray có MAX phần tử
    int nSize; // nSize là số phần tử được sử dụng, do user nhập

    // nhập mảng myArray tự động
    nhapMang(myArray, nSize);

    // xuất mảng myArray
    cout << "myArray: " << endl;
    xuatMang(myArray, nSize);

    int myArray2[MAX]; // mảng myArray2 có MAX phần tử

    // sao chép mảng myArray sang myArray2
    saoChepMangMotChieu(myArray2, myArray, nSize);

    // xuất mảng myArray2 sau khi sao chép
    cout << "myArray2: " << endl;
    xuatMang(myArray2, nSize);

    return 0;
}

// hàm nhập mảng
void nhapMang(int arr[], int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

    cout << "Nhap so luong phan tu n : ";
    cin >> n;

    // khởi tạo ngẫu nhiên từng phần tử từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand();
    }
}
```

```
}

// hàm xuất mảng
void xuatMang(int arr[], int n)
{
    // xuất từng phần tử cho mảng từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        cout << "array[" << i << "] = " << arr[i] << endl;
    }
}

// sao chép mảng nguồn sang mảng đích
void saoChepMangMotChieu(int arrDest[], int arrSource[], int n)
{
    for (int i = 0; i < n; i++)
    {
        arrDest[i] = arrSource[i];
    }
}
```

Output:



Tìm kiếm phần tử trong mảng

Yêu cầu: Tìm xem phần tử **x** có nằm trong mảng **myArray** kích thước **n** hay không? Nếu có thì nó nằm ở **vị trí đầu tiên nào?**

Ý tưởng: Xét từng phần của mảng **myArray**. Nếu phần tử đang xét bằng **x** thì trả về vị trí đó. Nếu không tìm được thì trả về **-1**.

Chương trình:

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
#include <string>
using namespace std;

// định nghĩa số phần tử mảng
#define MAX 1000

// khai báo prototype
void nhapMang(int arr[], int &n);
void xuatMang(int arr[], int n);
int timKiemPhanTuDauTien(int arr[], int n, int x);

int main()
{
    int myArray[MAX]; // mảng myArray có MAX phần tử
    int nSize; // nSize là số phần tử được sử dụng, do user nhập

    // nhập mảng myArray tự động
    nhapMang(myArray, nSize);

    // xuất mảng myArray
    cout << "myArray: " << endl;
    xuatMang(myArray, nSize);

    int x;
    cout << "Nhập phần tử x cần tìm: ";
    cin >> x;

    // tìm kiếm phần tử x đầu tiên trong mảng
    int idx = timKiemPhanTuDauTien(myArray, nSize, x);
    if (idx != -1)
        cout << "x nằm tại vị trí thứ " << idx << endl;

    return 0;
}
```

```
// hàm nhập mảng
void nhapMang(int arr[], int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

    cout << "Nhap so luong phan tu n : ";
    cin >> n;

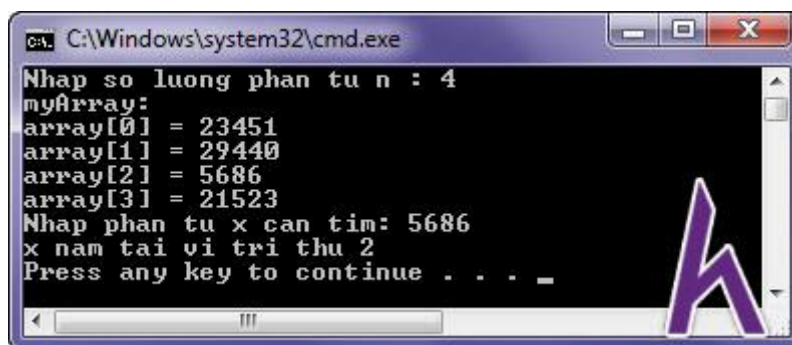
    // khởi tạo ngẫu nhiên từng phần tử từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand();
    }
}

// hàm xuất mảng
void xuatMang(int arr[], int n)
{
    // xuất từng phần tử cho mảng từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        cout << "array[" << i << "] = " << arr[i] << endl;
    }
}

// tìm kiếm phần tử x đầu tiên trong mảng
int timKiemPhanTuDauTien(int arr[], int n, int x)
{
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == x)
            return i;
    }

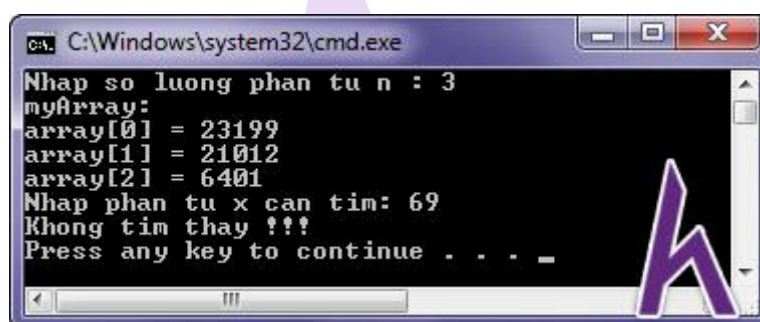
    return -1;
}
```

Output 1:



```
C:\Windows\system32\cmd.exe
Nhap so luong phan tu n : 4
myArray:
array[0] = 23451
array[1] = 29440
array[2] = 5686
array[3] = 21523
Nhap phan tu x can tim: 5686
x nam tai vi tri thu 2
Press any key to continue . . . _
```

Output 2:



```
C:\Windows\system32\cmd.exe
Nhap so luong phan tu n : 3
myArray:
array[0] = 23199
array[1] = 21012
array[2] = 6401
Nhap phan tu x can tim: 69
Khong tim thay !!!
Press any key to continue . . . _
```

Sắp xếp mảng 1 chiều

Yêu cầu: Cho trước mảng **myArray** kích thước **n**. Hãy sắp xếp mảng **a** đó sao cho các phần tử có giá trị tăng dần.

Ý tưởng: Sử dụng 2 biến **i** và **j** để so sánh tất cả cặp phần tử với nhau và hoán vị các cặp **ngược thế** (sai thứ tự).

Chương trình:

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
#include <string>
using namespace std;

// định nghĩa số phần tử mảng
#define MAX 1000

// khai báo prototype
void nhapMang(int arr[], int &n);
```

```
void xuatMang(int arr[], int n);
void hoanVi(int &a, int &b);
void sapXepTang(int arr[], int n);

int main()
{
    int myArray[MAX]; // mảng myArray có MAX phần tử
    int nSize; // nSize là số phần tử được sử dụng, do user nhập

    // nhập mảng myArray tự động
    nhapMang(myArray, nSize);

    // xuất mảng myArray
    cout << "myArray: " << endl;
    xuatMang(myArray, nSize);

    // sắp xếp mảng tăng
    sapXepTang(myArray, nSize);

    return 0;
}

// hàm nhập mảng
void nhapMang(int arr[], int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

    cout << "Nhap so luong phan tu n : ";
    cin >> n;

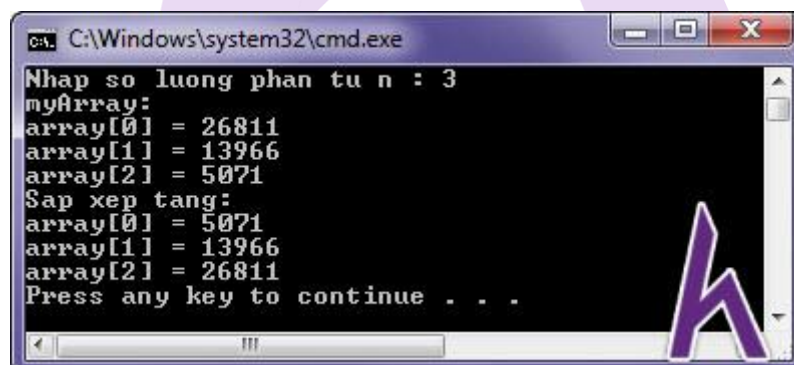
    // khởi tạo ngẫu nhiên từng phần tử từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand();
    }
}

// hàm xuất mảng
void xuatMang(int arr[], int n)
{
    // xuất từng phần tử cho mảng từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
```

```
{
    cout << "array[" << i << "] = " << arr[i] << endl;
}

// hoán vị giá trị 2 biến số
void hoanVi(int &a, int &b)
{
    int temp = a;
    a = b;
    b = temp;
}

// sắp xếp mảng tăng dần bằng thuật toán interchange sort
void sapXepTang(int arr[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            if (arr[i] > arr[j])
                hoanVi(arr[i], arr[j]);
        }
    }
}
```

Output:

Thêm và xóa một phần tử trong mảng

Thêm một phần tử vào mảng

Yêu cầu: Thêm phần tử **x** vào mảng **myArray** kích thước **n** tại vị trí **idx**.

Ý tưởng:

- “**Đẩy**” các phần tử bắt đầu tại vị trí **idx** sang phải **1 vị trí**.
- Đưa **x** vào vị trí **idx** trong mảng.
- Tăng **n** lên **1 đơn vị**.

Chương trình:

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
#include <string>
using namespace std;

// định nghĩa số phần tử mảng
#define MAX 1000

// khai báo prototype
void nhapMang(int arr[], int &n);
void xuatMang(int arr[], int n);
void themMotPhanTuVaoMang(int a[], int &n, int idx, int x);

int main()
{
    int myArray[MAX]; // mảng myArray có MAX phần tử
    int nSize; // nSize là số phần tử được sử dụng, do user nhập

    // nhập mảng myArray tự động
    nhapMang(myArray, nSize);

    // xuất mảng myArray
    cout << "myArray: " << endl;
    xuatMang(myArray, nSize);
}
```

```
// thêm 1 phần tử vào mảng
int idx;
cout << "Nhap vi tri can them: ";
cin >> idx;

int x;
cout << "Nhap gia tri can them: ";
cin >> x;
themMotPhanTuVaoMang(myArray, nSize, idx, x);

// xuất mảng sau khi thêm
cout << "myArray: " << endl;
xuatMang(myArray, nSize);

return 0;
}

// hàm nhập mảng
void nhapMang(int arr[], int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

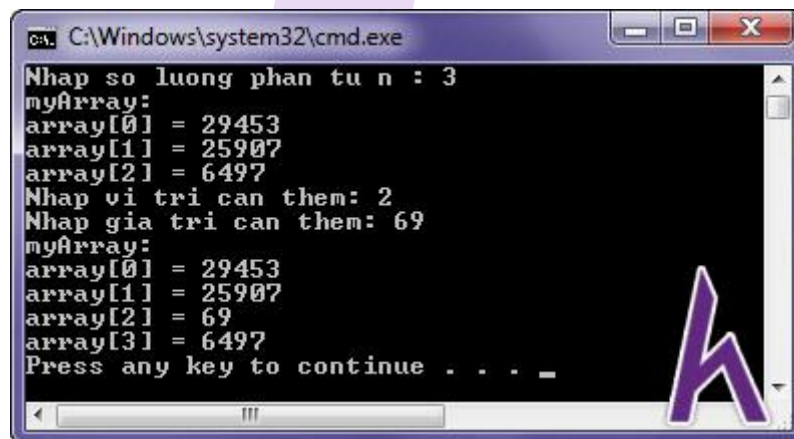
    cout << "Nhap so luong phan tu n : ";
    cin >> n;

    // khởi tạo ngẫu nhiên từng phần tử từ chỉ số 0 đến n – 1
    for (int i = 0; i < n; i++)
    {
        arr[i] = rand();
    }
}

// hàm xuất mảng
void xuatMang(int arr[], int n)
{
    // xuất từng phần tử cho mảng từ chỉ số 0 đến n – 1
    for (int i = 0; i < n; i++)
    {
        cout << "array[" << i << "] = " << arr[i] << endl;
    }
}
```

```
// thêm phần tử x vào mảng tại vị trí idx
void themMotPhanTuVaoMang(int a[], int &n, int idx, int x)
{
    if (idx >= 0 && idx <= n)
    {
        for (int i = n; i > idx; i--)
            a[i] = a[i - 1];
        a[idx] = x;
        n++;
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
Nhap so luong phan tu n : 3
myArray:
array[0] = 29453
array[1] = 25907
array[2] = 6497
Nhap vi tri can them: 2
Nhap gia tri can them: 69
myArray:
array[0] = 29453
array[1] = 25907
array[2] = 69
array[3] = 6497
Press any key to continue . . . _
```

Xóa một phần tử trong mảng

Yêu cầu: Xóa một phần tử trong mảng **a** kích thước **n** tại vị trí **vt**

Ý tưởng:

- “Kéo” các phần tử bên phải vị trí **idx** sang trái **1** vị trí.
- Giảm **n** xuống **1** đơn vị.

Chương trình:

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
#include <string>
using namespace std;

// định nghĩa số phần tử mảng
#define MAX 1000

// khai báo prototype
void nhapMang(int arr[], int &n);
void xuatMang(int arr[], int n);
void xoaMotPhanTuTrongMang(int a[], int &n, int idx);

int main()
{
    int myArray[MAX]; // mảng myArray có MAX phần tử
    int nSize; // nSize là số phần tử được sử dụng, do user nhập

    // nhập mảng myArray tự động
    nhapMang(myArray, nSize);

    // xuất mảng myArray
    cout << "myArray: " << endl;
    xuatMang(myArray, nSize);

    // xóa một phần tử trong mảng tại vị trí idx
    int idx;
    cout << "Nhập vị trí cần xóa: ";
    cin >> idx;

    xoaMotPhanTuTrongMang(myArray, nSize, idx);

    // xuất mảng sau khi xóa
    cout << "myArray: " << endl;
    xuatMang(myArray, nSize);

    return 0;
}

// hàm nhập mảng
void nhapMang(int arr[], int &n)
{

```

```
// khởi tạo số ngẫu nhiên
srand(time(NULL));

cout << "Nhap so luong phan tu n : ";
cin >> n;

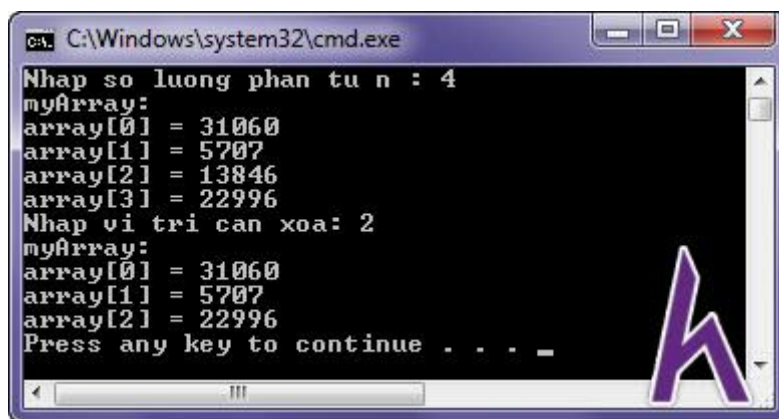
// khởi tạo ngẫu nhiên từng phần tử từ chỉ số 0 đến n - 1
for (int i = 0; i < n; i++)
{
    arr[i] = rand();
}

// hàm xuất mảng
void xuatMang(int arr[], int n)
{
    // xuất từng phần tử cho mảng từ chỉ số 0 đến n - 1
    for (int i = 0; i < n; i++)
    {
        cout << "array[" << i << "] = " << arr[i] << endl;
    }
}

// xóa một phần tử trong mảng tại vị trí idx
void xoaMotPhanTuTrongMang(int a[], int &n, int idx)
{
    if (idx >= 0 && idx < n)
    {
        for (int i = idx; i < n - 1; i++)
            a[i] = a[i + 1];

        n--;
    }
}
```

Output:



```
C:\Windows\system32\cmd.exe
Nhap so luong phan tu n : 4
myArray:
array[0] = 31060
array[1] = 5707
array[2] = 13846
array[3] = 22996
Nhap vi tri can xoa: 2
myArray:
array[0] = 31060
array[1] = 5707
array[2] = 22996
Press any key to continue . . .
```

Kết luận

Qua bài học này, bạn đã biết được [Các thao tác trên Mảng một chiều](#) trong C++. Còn rất nhiều thao tác khác trên mảng 1 chiều, trong phạm vi bài học không thể đề cập hết được, các bạn hãy tự mình tìm hiểu và bình luận bên dưới để chia sẻ cho mọi người nhé.

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn về [MẢNG HAI CHIỀU \(Multidimensional arrays\)](#) trong C++.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.