

# Bài 52. Thêm/ xóa phần tử trong mảng 1 chiều

Bởi Nguyễn Văn Hiếu -

Bài số 50 trong 69 bài của khóa học [Học C Không Khó](#)

Hôm nay, **Lập trình không khó** sẽ cùng các bạn đi giải quyết bài tập “**Thêm phần tử trong mảng 1 chiều và xóa phần tử trong mảng 1 chiều**” nhé. Đây là 1 bài tập thực hành hơi nâng cao một chút. Nó thường gây ra những khó khăn nhất định cho các bạn lần đầu tiếp xúc. Chúng ta sẽ cùng nhau giải quyết bài toán thêm xóa phần tử trong mảng ngay bây giờ nhé.

Bài này yêu cầu bạn có kiến thức lập trình cơ bản về ngôn ngữ C hoặc C++ nhé. Nếu kiến thức này bạn chưa thực sự vững, hãy tham khảo khóa học lập trình C cơ bản của mình [trên Youtube tại đây nhé](#).

## NỘI DUNG BÀI VIẾT

1. Xây dựng các hàm nhập xuất mảng
2. Thêm phần tử vào mảng 1 chiều
3. Xóa phần tử trong mảng 1 chiều
4. Full code thêm xóa phần tử trong mảng 1 chiều

## Xây dựng các hàm nhập xuất mảng

Đầu tiên, bạn hãy xử lý xong phần nhập xuất mảng nhé. Chúng ta bắt buộc phải sử dụng nhập/ xuất mảng để sử dụng trong quá trình thêm xóa phần tử mảng.

Code hàm nhập và hàm xuất mảng:

```
0
1 void NhapMang(int a[], int n){
2     for(int i = 0; i < n; i++){
3         printf("Nhap so thu %d: ", i);
4         scanf("%d", &a[i]);
5     }
6 }
```

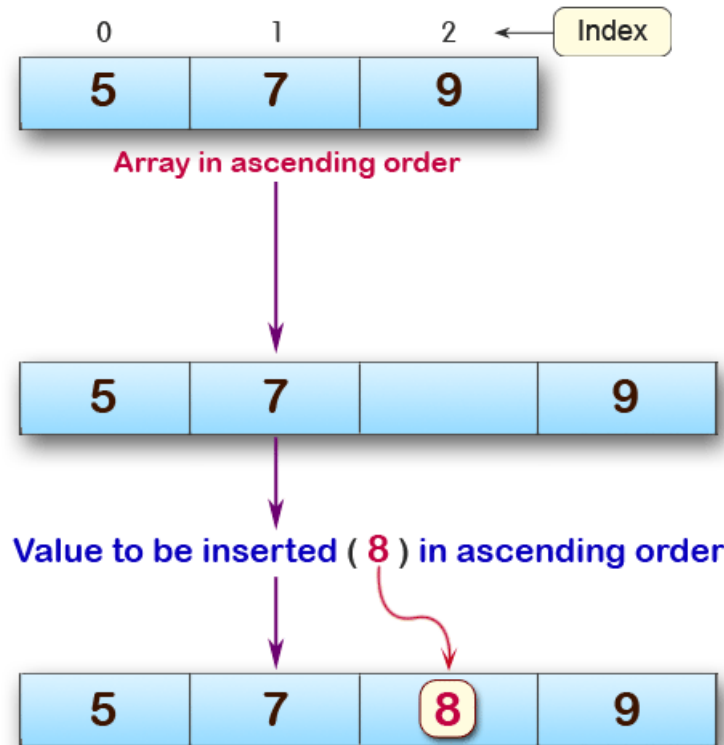
```
7
8 void XuatMang(int a[], int n){
9     for(int i = 0; i < n; i++){
10         printf("%4d", a[i]);
11     }
12 }
13
```

Bạn có thể xem hướng dẫn chi tiết phần này tại đây:

- [Hướng dẫn nhập xuất mảng 1 chiều](#)
- [Hướng dẫn nhập xuất mảng 2 chiều](#)

## Thêm phần tử vào mảng 1 chiều

Bạn có thể hình dung cách thêm 1 phần tử vào mảng qua hình ảnh dưới đây nhé.



© w3resource.com

*Cách thêm 1 phần tử mới vào mảng*

Như vậy, trước khi thực hiện thêm, bạn phải thực hiện lùi tất cả các số phía sau nơi được chèn(bao gồm cả vị trí chèn) về phía sau 1 ô nhớ. Như ảnh trên thì chỉ có số 9 bị ảnh hưởng, bị chuyển từ chỉ số 2 sang 3.

Sau khi dịch chuyển theo mô tả trên, chúng ta chỉ việc gán giá trị cần chèn cho phần tử tại nơi muốn chèn. Và cuối cùng, không quên tăng số lượng phần tử của mảng lên nhé.

Quy trình thêm phần tử vào mảng:

1. Kiểm tra mảng có thể thêm được phần tử nữa không? Nếu không, thoát hàm
2. Kiểm tra giá trị pos hợp lệ không. Ở đây nếu không hợp lệ mình cho về chỉ số đầu/cuối.
3. Thực hiện dịch chuyển mảng(phần phía sau nơi chèn + vị trí chèn)
4. Chèn vào vị trí cần chèn
5. Tăng số lượng phần tử

```
0
1 void ThemPhanTu(int a[], int &n, int val, int pos){
2     // Mảng đã đầy, không thể thêm.
3     if(n >= MAX){
4         return;
5     }
6     // Nếu pos <= 0 => Thêm vào đầu
7     if(pos < 0){
8         pos = 0;
9     }
10    // Nếu pos >= n => Thêm vào cuối
11    else if(pos > n){
12        pos = n;
13    }
14    // Dịch chuyển mảng để tạo ô trống trước khi thêm.
15    for(int i = n; i > pos; i--){
16        a[i] = a[i-1];
17    }
18    // Chèn val tại pos
19    a[pos] = val;
20    // Tăng số lượng phần tử sau khi chèn.
21    ++n;
22 }
23
```

## Xóa phần tử trong mảng 1 chiều

Bạn có thể hình dung cách xóa 1 phần tử trong mảng qua hình ảnh dưới đây nhé.

Arr[0]	Arr[1]	Arr[2]	Arr[3]	Arr[4]	Arr[5]	Arr[6]
45	23	32	72	67	56	12

↓  
After deleting element at 3<sup>rd</sup> position

Arr[0]	Arr[1]	Arr[2]	Arr[3]	Arr[4]	Arr[5]
45	23	32	67	56	12

Dễ nhận thấy, số lượng phần tử sẽ giảm đi 1. Tuy nhiên, hãy để ý các số từ 67 trở về sau đều bị giảm chỉ số đi 1 đơn vị.

Quy trình xóa phần tử trong mảng:

1. Kiểm tra có thể xóa hay không? Nếu không => thoát hàm
2. Kiểm tra giá trị pos hợp lệ không. Ở đây nếu không hợp lệ mình cho về chỉ số đầu/cuối.
3. Dịch chuyển mảng lùi 1 chỉ số – phần phía sau nơi xóa
4. Giảm số lượng phần tử

```

0
1 void XoaPhanTu(int a[], int &n, int pos){
2     // Mảng rỗng, không thể xóa.
3     if(n <= 0){
4         return;
5     }
6     // Nếu pos <= 0 => Xóa đầu
7     if(pos < 0){
8         pos = 0;
9     }
10    // Nếu pos >= n => Xóa cuối
11    else if(pos >= n){
12        pos = n-1;
13    }
14    // Dịch chuyển mảng
15    for(int i = pos; i < n - 1; i++){

```

```
16     a[i] = a[i+1];
17 }
18 // Giảm số lượng phần tử sau khi xóa.
19 --n;
20 }
21
```

**Gợi ý cho bạn:** Nếu bạn là người đam mê **thuật toán** và thích giải các bài tập như thế này, hãy tham khảo chuyên mục **bài tập c++** nhé.

## Full code thêm xóa phần tử trong mảng 1 chiều

```
0
1 // Thêm phần tử vào mảng 1 chiều
2
3 #include <stdio.h>
4 #define MAX 1000
5
6 void NhapMang(int a[], int n){
7     for(int i = 0; i < n; i++){
8         printf("Nhập số thứ %d: ", i);
9         scanf("%d", &a[i]);
10    }
11 }
12
13 void XuatMang(int a[], int n){
14     for(int i = 0; i < n; i++){
15         printf("%4d", a[i]);
16     }
17 }
18
19 void ThemPhanTu(int a[], int &n, int val, int pos){
20     // Mảng đã đầy, không thể thêm.
21     if(n >= MAX){
22         return;
23     }
24     // Nếu pos <= 0 => Thêm vào đầu
25     if(pos < 0){
26         pos = 0;
27     }
28     // Nếu pos >= n => Thêm vào cuối
29     else if(pos > n){
30         pos = n;
```

```

31     }
32     // Dịch chuyển mảng để tạo ô trống trước khi thêm.
33     for(int i = n; i > pos; i--){
34         a[i] = a[i-1];
35     }
36     // Thêm val tại pos
37     a[pos] = val;
38     // Tăng số lượng phần tử sau khi thêm.
39     ++n;
40 }
41
42 void XoaPhanTu(int a[], int &n, int pos){
43     // Mảng rỗng, không thể xóa.
44     if(n <= 0){
45         return;
46     }
47     // Nếu pos <= 0 => Xóa đầu
48     if(pos < 0){
49         pos = 0;
50     }
51     // Nếu pos >= n => Xóa cuối
52     else if(pos >= n){
53         pos = n-1;
54     }
55     // Dịch chuyển mảng
56     for(int i = pos; i < n - 1; i++){
57         a[i] = a[i+1];
58     }
59     // Giảm số lượng phần tử sau khi xóa.
60     --n;
61 }
62
63 int main(){
64     int a[MAX];
65     int n;
66     printf("\nNhập số lượng phần tử: ");
67     scanf("%d", &n);
68    NhapMang(a, n);
69    XuatMang(a, n);
70     printf("\n=====THEM PHAN TU=====\\n");
71     int val, pos;
72     printf("\nNhập số cần thêm: ");
73     scanf("%d", &val);
74     printf("\nNhập vị trí muốn thêm: ");

```

```

75     scanf("%d", &pos);
76     ThemPhanTu(a, n, val, pos);
77     printf("\nMang sau khi them: ");
78     XuatMang(a, n);
79     printf("\n=====XOA PHAN TU=====\\n");
80     printf("\\nNhap vi tri muon xoa: ");
81     scanf("%d", &pos);
82     XoaPhanTu(a, n, pos);
83     printf("\\nMang sau khi xoa: ");
84     XuatMang(a, n);
85     printf("\\nDone!");
86 }
87

```

Kết quả chạy thử:

```

0
1 hieunv@hieunv:~/projects$ g++ ThemXoaMang.cpp -o ThemXoaMang
2 hieunv@hieunv:~/projects$ ./ThemXoaMang
3
4 Nhap so luong phan tu: 5
5 Nhap so thu 0: 1
6 Nhap so thu 1: 2
7 Nhap so thu 2: 3
8 Nhap so thu 3: 4
9 Nhap so thu 4: 5
10 1 2 3 4 5
11 =====THEM PHAN TU=====
12
13 Nhap so can them: 6
14
15 Nhap vi tri muon chen: 1
16
17 Mang sau khi them: 1 6 2 3 4 5
18 =====XOA PHAN TU=====
19
20 Nhap vi tri muon xoa: 1
21
22 Mang sau khi xoa: 1 2 3 4 5
23

```



Bài trước: Bài 51. Tìm số lớn thứ 2 trong mảng

Bài sau: Bài 53. Bài tập mảng 1 chiều có lời giải

Nguyễn Văn Hiếu

Sáng lập cộng đồng Lập Trình Không Khó với mong muốn giúp đỡ các bạn trẻ trên con đường trở thành những lập trình viên tương lai. Tất cả những gì tôi viết ra đây chỉ đơn giản là sở thích ghi lại các kiến thức mà tôi tích lũy được.



BÀI VIẾT HAY

CHUYÊN MỤC HAY

Blog chia sẻ kiến thức lập trình của Hiếu, xây dựng cộng đồng những người học lập trình. Cho đi kiến thức mình có là cách học tập hiệu quả nhất

👉 Báo lỗi / Liên hệ / Hợp tác / Quảng cáo

Bài 1. Giới thiệu khóa học “Học C Bá Đạo”  
21/07/2019

1000 bài tập lập trình C/C++ có lời giải của thầy Khang  
25/12/2019

Kiểm tra số nguyên tố sử dụng C/C++ và Java  
15/07/2018

Học C/C++	194
Học Python	48
Học Java	45
Học Javascript	37
Khóa học	33
Học Web	26
Học C#	24
Chia sẻ	23





### - BẠN BÈ & ĐỐI TÁC -

Luyện Code - Tự Học Đồ Họa - Cách Học Lập Trình - VNTALKING

© 2018-2020. Bản quyền thuộc Lập Trình Không Khó. [Privacy & Terms](#)

