



Lập trình C/C++

Các kỹ thuật trên mảng một chiều trong C/C++

2017/07/25 • 13.378 lượt xem • by Minh Hoàng • 2 bình luận • 12 phút đọc



C PROGRAMMING

Các kỹ thuật thao tác với
mảng một chiều



2

Được viết bởi Minh Hoàng

Series lập trình C/C++, ngôn ngữ lập trình hệ thống mạnh mẽ.

- Trong nội dung của bài viết này, mình sẽ trình bày một số kỹ thuật quan trọng khi thao tác với mảng 1 chiều trong lập trình C. Vì bài viết có liên quan đến kiến thức của 3 phần bên dưới nên các bạn có thể xem lại nếu cần nhé:

1. Hàm và cách sử dụng trong lập trình C/C++
2. Tham trị, tham chiếu trong lập trình C/C++
3. Mảng một chiều trong lập trình C

- Nội dung sẽ trình bày các vấn đề sau:

1. Nhập, xuất mảng các số nguyên.
2. Kiểm tra mảng có phải toàn số dương hay không?
3. Tìm giá trị lớn nhất, nhỏ nhất của mảng.
4. Sắp xếp mảng tăng dần, giảm dần.
5. Tách mảng (tách các số âm trong mảng ra một mảng mới)
 - Mảng ban đầu: 8, -5, 0, -2, 9
 - Tách thành 1 mảng mới: -5, -2
6. Gộp mảng (gộp mảng vừa tách vào mảng cũ)
 - Gộp lại thành: 8, -5, 0, -2, 9, -5, -2
7. Tìm kiếm một phần tử trong mảng.
8. Thêm một phần tử vào mảng.
9. Xóa một phần tử khỏi mảng.
10. Hàm main thực hiện chương trình.

1. Nhập, xuất mảng các số nguyên. 


Home	C# ▾	C/C++ ▾	VB.Net ▾	JS ▾	IT ▾	Download ▾	Japan Life ▾	Sitemap	Welcome!	Login
1	/*									
2	* Hàm nhập mảng: tham số là mảng kiểu nguyên, và số phần tử n của mảng sẽ									
3	* truyền kiểu tham chiếu (vì muốn sau khi ra khỏi hàm nhập thì giá trị n									
4	* cũng sẽ dc mang theo đến hàm main để dùng thực hiện các chức năng khác)									
5	*/									
6	void NhapMang(int a[], int &n)									
7	{									
8	// Nhập số phần tử của mảng và kiểm tra tính hợp lệ									
9	do {									
10	printf("\nNhap so luong phan tu cua mang: ");									
11	scanf_s("%d", &n);									
12										
13	if (n <= 0 n > MAX) {									
14	printf("So luong phan tu cua mang khong hop le (1 -> %d).\n", MAX);									
15	}									
16	} while (n <= 0 n > MAX);									
17										
18	// Nhập giá trị cho mảng									
19	for (int ii= 0; ii < n; ii++)									
20	{									
21	printf("\nNhap a[%d] = ", ii);									
22	scanf_s("%d", &a[ii]);									
23	}									
24	}									
25										
26	/*									
27	* Hàm xuất mảng: tham số là mảng kiểu nguyên, và số phần tử n của mảng									
28	* truyền kiểu tham chiếu HAY tham trị đều được (vì đối với hàm xuất không									
29	* làm thay đổi giá trị của n, nên sẽ truyền n theo kiểu tham trị)									
30	*/									
31	void XuatMang(int a[], int n)									
32	{									
33	for (int ii = 0; ii < n; ii++)									
34	{									
35	printf("%d ", a[ii]);									
36	}									
37	}									

2. Kiểm tra mảng có phải toàn số dương hay không? 

2. Kiểm tra mảng có phải toàn số dương hay không?

```

1  /*
2  * Hàm kiểm tra mảng có phải là toàn số dương hay không
3  */
4  bool KiemtraMang(int a[], int n)
5  {
6      for (int ii = 0; ii < n; ii++)
7      {
8          if (0 > a[ii]) {
9              // Nếu có số âm thì trả về kết quả luôn, không cần kiểm tra nữa.
10             return false;
11         }
12     }
13     return true;
14 }
15 
```

3. Tìm giá trị lớn nhất, nhỏ nhất của mảng. 

Home	C# ▾	C/C++ ▾	VB.Net ▾	JS ▾	IT ▾	Download ▾	Japan Life ▾	Sitemap	Welcome!	Login
------	------	---------	----------	------	------	------------	--------------	---------	----------	-------

```

1  /*
2  * Hàm tìm giá trị nhỏ nhất trong mảng
3  */
4  int TimMin(int a[], int n)
5  {
6      // Cho min bằng phần tử đầu tiên của mảng
7      int min = a[0];
8
9      // Từ đó check min với phần tử thứ 2 đến hết mảng
10     for (int ii = 1; ii < n; ii++)
11     {
12         if (a[ii] < min) {
13             min = a[ii];
14         }
15     }
16
17     return min;
18 }
19
20 /*
21 * Hàm tìm giá trị lớn nhất trong mảng
22 */
23 int TimMax(int a[], int n)
24 {
25     int max = a[0];
26
27     for (int ii = 1; ii < n; ii++)
28     {
29         if (a[ii] > max) {
30             max = a[ii];
31         }
32     }
33
34     return max;
35 }

```

4. Sắp xếp mảng tăng dần, giảm dần. ➡

Home	C# ▾	C/C++ ▾	VB.Net ▾	JS ▾	IT ▾	Download ▾	Japan Life ▾	Sitemap	Welcome!	Login
1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39					

```

/*
 * Hàm hoán vị 2 số nguyên
 */
void HoanVi(int &x, int &y)
{
    int tmp = x;
    x = y;
    y = tmp;
}

/*
 * Hàm sắp xếp mảng
 */
void SapXepMang(int a[], int n, bool isTangDan)
{
    // Thực hiện so sánh tuần tự
    for (int ii = 0; ii < n - 1; ii++)
    {
        for (int jj = ii + 1; jj < n; jj++)
        {
            // Nếu là sắp xếp tăng dần
            if (isTangDan)
            {
                if (a[ii] > a[jj])
                {
                    HoanVi(a[ii], a[jj]);
                }
            }
            // Nếu là sắp xếp giảm dần
            else
            {
                if (a[ii] < a[jj])
                {
                    HoanVi(a[ii], a[jj]);
                }
            }
        }
    }
}

```

5. Tách mảng (tách các số âm trong mảng ra một mảng mới) 

5. Tách mảng (tách các số âm trong mảng ra một mảng mới).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----

```

/*
 * Tách mảng (Vì số phần tử của [mảng Temp] sau khi tách sẽ thay đổi,
 * nên sẽ truyền kiểu tham chiếu)
 */
void TachMang(int a[], int n, int Temp[], int &nTemp)
{
    nTemp = 0;

    // Duyệt mảng a để tìm ra các số âm lưu vào mảng Temp
    for (int ii = 0; ii < n; ii++)
    {
        if (0 > a[ii])
        {
            Temp[nTemp++] = a[ii];

            /* Tương đương với cách viết:
            Temp[nTemp] = a[ii];
            nTemp++;
            */
        }
    }
}

```

6. Gộp mảng (gộp mảng vừa tách vào mảng cũ) 

2

Home	C# ▾	C/C++ ▾	VB.Net ▾	JS ▾	IT ▾	Download ▾	Japan Life ▾	Sitemap	Welcome!	Login
1	/*									
2	* Gộp mảng (Vì số phần tử của [mảng a] sau khi gộp sẽ thay đổi,									
3	* nên sẽ truyền kiểu tham chiếu)									
4	*/									
5	void GopMang(int a[], int &n, int Temp[], int nTemp)									
6	{									
7	// Giữ lại số phần tử ban đầu của mảng a									
8	int m = n;									
9										
10	// Gộp số lượng phần tử của mảng Temp vào mảng a									
11	n += nTemp;									
12										
13	// Thêm các phần tử của mảng Temp vào mảng a									
14	for (int ii = (n - nTemp); ii < n; ii++)									
15	{									
16	a[ii] = Temp[ii - m];									
17	}									
18										
19	/* a : 1, 2, 3, 4, 5									
20	Temp : 6, 7									
21	a[5] = Temp[0]									
22	a[6] = Temp[1] */									
23	}									

7. Tìm kiếm một phần tử trong mảng. 

7. Tìm kiếm một phần tử trong mảng.

```

1  /*
2  * Hàm tìm kiếm xem 1 số có trong mảng hay không
3  */
4  bool Timkiem(int a[], int n, int x) // x là phần tử cần tìm kiếm
5  {
6      // Duyệt từng phần tử trong mảng
7      for (int ii = 0; ii < n; ii++) {
8          // đem so sánh với x
9          if (x == a[ii])
10             {
11                 return true;
12             }
13     }
14     return false;
15 }
16

```

8. Thêm một phần tử vào mảng. 

Home	C# ▾	C/C++ ▾	VB.Net ▾	JS ▾	IT ▾	Download ▾	Japan Life ▾	Sitemap	Welcome!	Login
1	/* Hàm thêm 1 phần tử vào 1 vị trí trong mảng									
2	*/									
3										
4	void ThemPhanTu(int a[], int &n, int PhanTuThem, int ViTriThem)									
5	{									
6	/* Mảng a ban đầu:									
7	· chỉ số: 0 1 2 3 4 5 6 7									
8	· phần tử: a b c d e f g h									
9										
10	Thêm phần tử X vào vị trí thứ 3 trong mảng.									
11										
12	Mảng a sau khi thêm:									
13	· chỉ số: 0 1 2 3 4 5 6 7 8									
14	· phần tử: a b c X d e f g h									
15										
16	VỊ TRÍ SAU = VỊ TRÍ TRƯỚC									
17	a[4] = a[3]									
18	a[5] = a[4]									
19	a[6] = a[5]									
20	a[7] = a[6]									
21	a[8] = a[7]									
22										
23	Nếu for() từ trên xuống thì tất cả giá trị sẽ là của a[3]									
24	(vì qua các lần gán kết quả bị ghi đè)									
25	Do đó, sẽ thực hiện for() từ <<dưới lên>>, rồi thực hiện phép gán giá trị.									
26										
27	* Cách 1: Chọn theo VỊ TRÍ SAU, ta thấy:									
28	· a[4] = (Vị trí Thêm + 1) = 3 + 1									
29	· a[8] = Tổng số phần tử ban đầu của mảng a = 8									
30	for (int ii = n; ii >= ViTriThem + 1; ii--)									
31	{									
32	//SAU = TRƯỚC									
33	a[ii] = a[ii - 1];									
34	}									
35										
36	* Cách 2: Chọn theo VỊ TRÍ TRƯỚC, ta thấy:									
37	· a[3] = Vị trí Thêm = 3									
38	· a[7] = (Tổng số phần tử ban đầu của mảng a - 1) = 8 - 1									
39	for (int ii = n - 1; ii >= ViTriThem; ii--)									
40	{									
41	//SAU = TRƯỚC									
42	a[ii + 1] = a[ii];									
43	}									
44	*/									
45	for (int ii = n; ii >= ViTriThem + 1; ii--)									
46	{									
47	//SAU = TRƯỚC									
48	a[ii] = a[ii - 1];									
49	}									
50										
51	// Tăng số lượng phần tử của mảng a									
52	n++;									
53										
54	a[ViTriThem] = PhanTuThem;									
55	}									

9. Xóa một phần tử khỏi mảng. ➡

Home	C# ▾	C/C++ ▾	VB.Net ▾	JS ▾	IT ▾	Download ▾	Japan Life ▾	Sitemap	Welcome!	Login
1	/* Hàm xóa 1 phần tử tại 1 vị trí trong mảng									
2	*/									
3	void XoaPhanTu(int a[], int &n, int ViTriXoa)									
4	{									
5	/* Mảng a ban đầu:									
6	· chỉ số: 0 1 2 3 4 5 6 7									
7	· phần tử: a b c d e f g h									
8										
9	Xóa phần tử X tại vị trí thứ 3 trong mảng.									
10										
11	Mảng a sau khi xóa:									
12	· chỉ số: 0 1 2 3 4 5 6									
13	· phần tử: a b c e f g h									
14										
15	VỊ TRÍ SAU = VỊ TRÍ TRƯỚC									
16	a[3] = a[4]									
17	a[4] = a[5]									
18	a[5] = a[6]									
19	a[6] = a[7]									
20										
21										
22	Nếu for() từ dưới lên, thì tất cả giá trị sẽ là của a[7]									
23	(vì qua các lần gán kết quả bị ghi đè))									
24	Do đó, sẽ thực hiện for() từ <<trên xuống>>, rồi thực hiện phép gán giá trị.									
25										
26	* Cách 1: Chọn theo VỊ TRÍ SAU, ta thấy:									
27	· a[3] = Vị trí xóa = 3									
28	· a[6] = Ban đầu vị trí cuối mảng là (n - 1), giờ xóa đi 1 phần tử thì vị trí cuối mảng									
29	//OR: for (int ii = ViTriXoa; ii < (n - 1); ii++)									
30	for (int ii = ViTriXoa; ii <= (n - 2); ii++)									
31	{									
32	//SAU = TRƯỚC									
33	a[ii] = a[ii + 1];									
34	}									
35										
36	* Cách 2: Chọn theo VỊ TRÍ TRƯỚC, ta thấy:									
37	· a[4] = (Vị trí Thêm + 1) = 3 + 1									
38	· a[7] = (Tổng số phần tử ban đầu của mảng a - 1) = 8 - 1									
39	// OR for (int ii = ViTriXoa + 1; ii < n; ii++)									
40	for (int ii = ViTriXoa + 1; ii <= n - 1; ii++)									
41	{									
42	//SAU = TRƯỚC									
43	a[ii - 1] = a[ii];									
44	}									
45	*/									
46	for (int ii = ViTriXoa + 1; ii <= n - 1; ii++)									
47	{									
48	//SAU = TRƯỚC									
49	a[ii - 1] = a[ii];									
50	}									
51										
52	// Giảm số lượng phần tử của mảng a									
53	n--;									
54	}									

10. Hàm main thực hiện chương trình

10. Hàm main thực hiện chương trình.

```

1  #include <stdio.h>
2  #include <iostream>
3
4  // Định nghĩa số phần tử lớn nhất của mảng
5  #define MAX 50
6
7  // Đặt hàm cần thực hiện chức năng cụ thể vào đây.
8
9  int main()
10 {
11     // Khai báo mảng
12     int a[MAX];
13
14     // Khai báo số lượng phần tử của mảng
15     int n;
16
17     // Nhập

```


Nhap so luong phan tu cua mang: 6

Nhap a[0] = 5

Nhap a[1] = -2

Nhap a[2] = 9

Nhap a[3] = 1

Nhap a[4] = 0

Nhap a[5] = -4

Mang vua nhap la: 5 -2 9 1 0 -4

Mang co ton tai so am.

Phan tu nho nhat trong mang la: -4

Phan tu lon nhat trong mang la: 9

Mang sap xep tang dan: -4 -2 0 1 5 9

Mang sap xep giam dan: 9 5 1 0 -2 -4

Mang duoc tach: -2 -4

Mang sau khi gop: 9 5 1 0 -2 -4 -2 -4

Nhap vao so muon tim kiem: 7

Khong tim thay so 7 trong mang.

Nhap vi tri them (0 -> 8): 5

Nhap phan tu them: 7

Mang sau khi them phan tu 7 vao vi tri 5 la: 9 5 1 0 -2 7 -4 -2 -4

Nhap vi tri xoa (0 -> 8): 6

Mang sau khi xoa phan tu tai vi tri 6 la: 9 5 1 0 -2 7 -2 -4

Kết quả chương trình

Cảm ơn bạn đã theo dõi. Đừng ngần ngại hãy cùng thảo luận với chúng tôi!

5

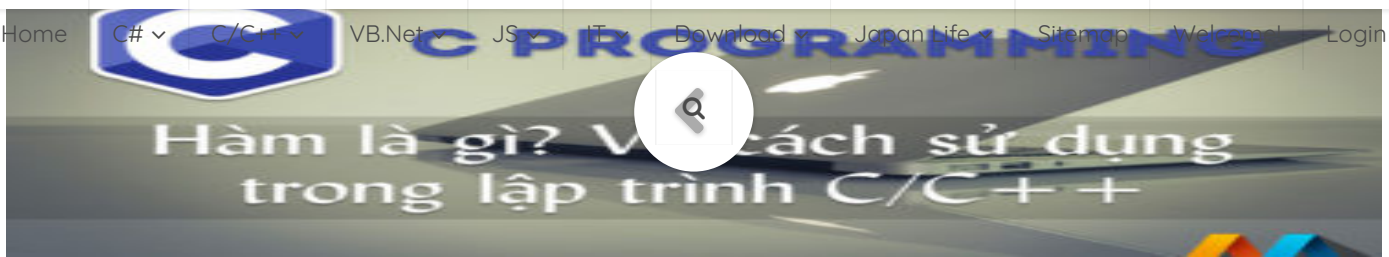
Article Rating



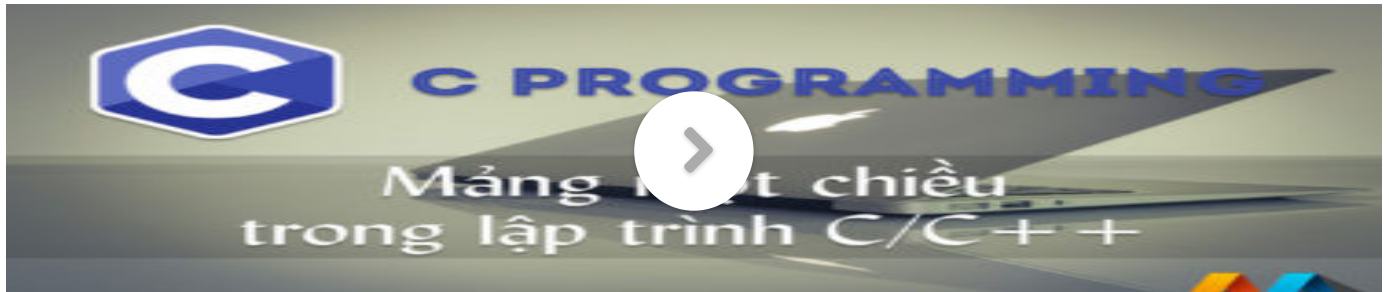
#array #c programming #lập trình #mảng #mảng 1 chiều #mảng một chiều



2



Hàm là gì? Và cách sử dụng trong lập trình C/C++



Mảng một chiều trong lập trình C/C++

BÀI VIẾT NỔI BẬT CÙNG CHUYÊN MỤC



Lập trình C/C++
Hàm memset() trong
C/C++



Lập trình C/C++
Hàm memset() trong
C/C++



Lập trình C/C++
Danh sách định dạng
nhập - xuất dữ liệu...



Lập trình C/C++
Nguyên mẫu hàm trong
C/C++ là gì? Và cách...



Lập trình C/C++
Kỹ thuật dùng con trỏ
thao tác với mảng một chiều



Lập trình C/C++
Con trỏ hằng và Hằng
con trỏ trong C/C++

GIỚI THIỆU



Minh Hoàng

Xin chào, tôi là Hoàng Ngọc Minh, hiện đang làm BrSE, tại công ty Toyota, Nhật Bản. Những gì tôi viết trên blog này là những trải nghiệm thực tế tôi đã đúc rút ra được trong cuộc sống, quá trình học tập và làm việc. Các bài viết được biên tập một cách chi tiết, linh hoạt để giúp bạn đọc có thể tiếp cận một cách dễ dàng nhất. Hi vọng nó sẽ có ích hoặc mang lại một góc nhìn khác cho bạn[...]

2

Xem tất cả bài viết

B

I

U

☒

☰

☰

”

⌂


☒

{ }

[+]



2 COMMENTS

  Thích nhiều nhất ▾



Mảng 1 chiều trong lập trình C - Minh Hoàng Blog | Cùng nhau chia sẻ kiến thức lập trình!

🕒 4 years ago

[...] phần như sau các bạn làm thử trước nhé, sau khi làm xong thì theo dõi bài viết Các kỹ thuật trên mảng 1 chiều mà mình đã trình bày. Rồi cùng comment trao đổi cách làm [...]

👍 0 👎 ➡ Reply



Kỹ thuật dùng con trỏ thao tác với mảng một chiều - Minh Hoàng Blog | Cùng nhau chia sẻ kiến thức lập trình!

🕒 4 years ago

[...] Ở bài viết các kỹ thuật trên mảng 1 chiều (không sử dụng con trỏ) đã giải thích chi tiết các kỹ thuật thao tác với [...]

👍 0 👎 ➡ Reply

Bài viết được xem nhiều

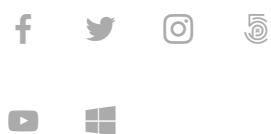
Bài viết được bình luận nhiều

Welcome to Minh Hoàng Blog

Welcome to Minh Hoàng











Kết nối với chúng tôi



Translate »



Home	C#	C/C++	VB.Net	JavaScript	Download	Japan Life	Sitemap	Welcome!	Login
		Cách kiểm tra thu nhập, thông tin của một kênh...		Hướng dẫn thủ tục bảo lãnh vợ/chồng, con sang Nhật					
		 <p>Các phím tắt thông dụng trong Visual Studio</p>		 <p>Hướng dẫn hủy hợp đồng sim Data 4G Rakuten</p>					
		 <p>Cách xuất video (render) nhẹ mà nét trong After...</p>		 <p>Hướng dẫn đăng ký thẻ credit visa của Rakuten</p>					
		 <p>Hướng dẫn thủ tục bảo lãnh vợ/chồng, con sang Nhật</p>		 <p>Hướng dẫn đăng ký sim giá rẻ Data 4G Rakuten</p>					
		 <p>Tính kế thừa trong lập trình C#</p>		 <p>Hướng dẫn điền form xin Visa Nhật Bản</p>					