Array algorithms in C++ STL (all_of, any_of, none_of, copy_n and iota)

From C++11 onwards, some new and interesting algorithms are added in STL of C++. These algorithms operate on an array and are useful in saving time during coding and hence useful in competitive programming as well.

all_of()

This function operates on whole range of array elements and can save time to run a loop to check each elements one by one. It checks for a given property on every element and returns true when each element in range satisfies specified property, else returns false.

```
// C++ code to demonstrate working of all_of()
#include<iostream>
#include<algorithm> // for all_of()
using namespace std;
int main()
{
    // Initializing array
    int ar[6] =  {1, 2, 3, 4, 5, -6};

    // Checking if all elements are positive
    all_of(ar, ar+6, [](int x) { return x>0; })?
          cout << "All are positive elements" :
          cout << "All are not positive elements";

    return 0;

}
```

Output:

```
All are not positive elements
```

In the above code, -6 being a negative element negates the condition and returns false.

any_of()

This function checks for a given range if there’s even one element satisfying a given property mentioned in function. Returns true if at least one element satisfies the property else returns false.

```
// C++ code to demonstrate working of any_of()
#include<iostream>
#include<algorithm> // for any_of()
using namespace std;
int main()
{
    // Initializing array
    int ar[6] =  {1, 2, 3, 4, 5, -6};

    // Checking if any element is negative
    any_of(ar, ar+6, [](int x){ return x<0; })?
          cout << "There exists a negative element" :
          cout << "All are positive elements";

    return 0;

}
```

Output:

```
There exists a negative element
```

In above code, -6 makes the condition positive.

none_of()

This function returns true if none of elements satisfies the given condition else returns false.

```
// C++ code to demonstrate working of none_of()
#include<iostream>
#include<algorithm> // for none_of()
using namespace std;
int main()
{
    // Initializing array
    int ar[6] =  {1, 2, 3, 4, 5, 6};

    // Checking if no element is negative
    none_of(ar, ar+6, [](int x){ return x<0; })?
          cout << "No negative elements" :
          cout << "There are negative elements";

    return 0;
}
```

Output:

```
No negative elements
```

Since all elements are positive, the function returns true.

copy_n()

copy_n() copies one array elements to new array. This type of copy creates a deep copy of array. This function takes 3 arguments, source array name, size of array and the target array name.

```
// C++ code to demonstrate working of copy_n()
#include<iostream>
#include<algorithm> // for copy_n()
using namespace std;
int main()
{
    // Initializing array
    int ar[6] =  {1, 2, 3, 4, 5, 6};

    // Declaring second array
    int ar1[6];

    // Using copy_n() to copy contents
    copy_n(ar, 6, ar1);

    // Displaying the copied array
    cout << "The new array after copying is : ";
    for (int i=0; i<6 ; i++)
       cout << ar1[i] << " ";

    return 0;

}
```

Output:

```
The new array after copying is : 1 2 3 4 5 6
```

In the above code, the elements of ar are copied in ar1 using copy_n()

iota()

This function is used to assign continuous values to array. This function accepts 3 arguments, the array name, size, and the starting number.

```
// C++ code to demonstrate working of iota()
#include<iostream>
#include<numeric> // for iota()
using namespace std;
int main()
{
    // Initializing array with 0 values
    int ar[6] =  {0};

    // Using iota() to assign values
    iota(ar, ar+6, 20);

    // Displaying the new array
    cout << "The new array after assigning values is : ";
    for (int i=0; i<6 ; i++)
       cout << ar[i] << " ";

    return 0;

}
```

Output:

```
The new array after assigning values is : 20 21 22 23 24 25
```

In the above code, continuous values are assigned to array using iota().

This article is contributed by **Manjeet Singh** .If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

std::iota in C++

numeric header in C++ STL | Set 2 (adjacent_difference(), inner_product() and iota())

Maximum subarray sum in array formed by repeating the given array k times

Maximum sub-array sum after dividing array into sub-arrays based on the given queries

Find the Initial Array from given array after range sum queries

Difference between pointer to an array and array of pointers

Maximize the sum of array by multiplying prefix of array with -1

Find the non decreasing order array from given array

array::rbegin() and array::rend() in C++ STL

Sorting an array according to another array using pair in STL

array::crbegin() and array::crend() in C++ STL

array::cbegin() and array::cend() in C++ STL

Array with GCD of any of its subset belongs to the given array

array::front() and array::back() in C++ STL

array::fill() and array::swap() in C++ STL

Improved By : mohitw16

Article Tags : C C++ Competitive Programming cpp-algorithm-library cpp-array CPP-Library STL

Practice Tags : STL C CPP

👍 18

2.4

□ To-do □ Done

Based on **41** vote(s)

Feedback/ Suggest Improvement   Add Notes   Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

GeeksforGeeks
A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Courses
Company-wise
Topic-wise
How to begin?

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

▲

@geeksforgeeks, Some rights reserved