

Основные итераторы

Итераторы вывода(Output iterators)

Итераторы вывода возвращает итератор потока вывода:

`ostream_iterator`

Итераторы вывода возвращают итераторы вставки:

`inserter, front_inserter, back_inserter`

Алгоритм merge()

```
template <class InputIterator1, class InputIterator2,
          class OutputIterator>
OutputIterator merge(InputIterator1 first1, InputIterator1 last1,
                    InputIterator2 first2, InputIterator2 last2,
                    OutputIterator result)
{
    while (first1 != last1 && first2 != last2)
    {
        if (*first2 < *first1)
            *result++ = *first2++;
        else
            *result++ = *first1++;
    }
    return copy(first2, last2, copy(first1, last1, result));
}
```

Алгоритм copy()

```
template <class InputIterator, class OutputIterator>
OutputIterator copy(InputIterator first, InputIterator last,
                   OutputIterator result)
{
    while (first != last) *result++ = *first++;
    return result;
}
```

```

#include <iostream>
#include <algorithm>
#include <vector>
#include <iterator>

int main()
{
    using namespace std;

    int a1[] = { 0, 1, 2, 3, 4 };
    int a2[] = { 5, 6, 7, 8, 9 };
    // Объявление стандартного контейнера - вектор элементов
    vector<int> v(10);

    // Слияние двух массивов в вектор
    merge(a1, a1 + 5, a2, a2 + 5, v.begin());

    // Копирование элементов вектора в поток вывода
    copy(v.begin(), v.end(), ostream_iterator<int>(cout));
    cout << endl;

    // Копирование элементов вектора в поток вывода
    copy(v.begin(), v.end(), ostream_iterator<int>(cout, " "));
    cout << endl;
    return 0;
}

```

Результат работы программы:

```

0123456789
0 1 2 3 4 5 6 7 8 9

```