

# reverse()

将容器 `[start, end)` 内所有元素翻转。

```
1  #include <iostream>
2  #include <cstdio>
3  #include <vector>
4  #include <algorithm>
5  using namespace std;
6
7  int main() {
8
9      vector<int> v({1, 2, 3, 4});
10
11     reverse(v.begin(), v.end());
12
13     for(auto x : v)
14         cout << x << " ";
15
16     return 0;
17 }
```

```
1  #include <iostream>
2  #include <cstdio>
3  #include <vector>
4  #include <algorithm>
5  using namespace std;
6
7  int main() {
8
9      int a[10] = {6, 7, 8, 9, 1, 2, 3, 5};
10     reverse(a + 0, a + 3 + 1);
11
12     for(auto x : a)
13         cout << x << " ";
14
15     return 0;
16 }
```

# unique

```
1  #include <iostream>
2  #include <cstdio>
3  #include <algorithm>
4  #include <vector>
5  using namespace std;
6
7  int main() {
```

```

8
9     vector<int> v({1, 1, 2, 2, 3, 4, 5});
10    v.erase(unique(v.begin(), v.end()), v.end());
11    /*
12    int p = unique(v.begin(), v.end()) - v.begin();
13    cout << p << endl;
14    */
15
16    /*
17    for(int i = 0; i < p; i++)
18        cout << v[i] << " ";
19    */
20
21    for(auto x : v)
22        cout << x << " ";
23
24
25    return 0;
26 }

```

```

1  #include <iostream>
2  #include <cstdio>
3  #include <algorithm>
4  using namespace std;
5
6  int main() {
7
8      int a[] = {1, 1, 2, 2, 3, 4, 5};
9
10     int p = unique(a, a + 7) - a;
11     for(int i = 0; i < p; i++)
12         cout << a[i] << " ";
13
14     return 0;
15 }

```

## random\_shuffle

```

1  #include <iostream>
2  #include <cstdio>
3  #include <algorithm>
4  #include <vector>
5  #include <ctime>
6  using namespace std;
7
8  int main() {
9
10     srand(time(0));
11
12     vector<int> v({1, 2, 3, 4, 5, 6});
13
14     random_shuffle(v.begin(), v.end());

```

```
15
16     for(int x : v)
17         cout << x << " ";
18
19     puts("");
20
21     sort(v.begin(), v.end(), greater<int>());
22
23     for(int x : v)
24         cout << x << " ";
25
26     puts("");
27
28     sort(v.begin(), v.end(), less<int>());
29
30     for(int x : v)
31         cout << x << " ";
32
33     puts("");
34
35     return 0;
36 }
```