

Контейнеры

pair и vector

vector — динамический массив

```
#include <vector>
using namespace std;
...
vector<int> a;
vector<int> x = {25,7,13};
vector<int> b(3);
a.push_back(2);
a.push_back(9);
a.insert(a.begin()+1,3);
for (int i=0; i<a.size(); ++i)
    printf("%d ",a[i]);
```

vector<bool> — побитовое сжатие

Контейнеры

pair и vector

vector — динамический массив

```
#include <vector>
using namespace std;
...
vector<int> a;
vector<int> x = {25,7,13};
vector<int> b(3);
a.push_back(2);
a.push_back(9);
a.insert(a.begin()+1,3);
for (int i=0; i<a.size(); ++i)
    printf("%d ",a[i]);
```

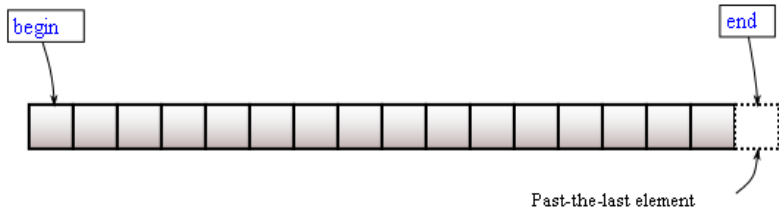
vector<bool> — побитовое сжатие

```
#include <utility>
using namespace std;
...
pair<int, char> a = make_pair(5, 'A');
a.first+=2;
a.second='Z';
printf("%d %c", a.first, a.second);
```

`a.begin()`

`a.end()`

— итераторы на начальный и следующий за конечным элемент



Алгоритмы стандартной библиотеки

```
#include <algorithm>
using namespace std;
vector<int> a;
int d[n];
bool cmp(int x, int y){ return (x<y); }
bool cmp2(int x, int y){ return (d[x]<d[y]); }
m = max_element(a.begin(), a.end());
m = min_element(a.begin(), a.end());
sum = accumulate(a.begin(), a.end(), 0);
product = accumulate(a.begin(), a.end(), 1,
    multiplies<int>());
sort(a.begin(), a.end());
sort(d, d+n);
sort(a.begin(), a.end(), cmp);
iota(a.begin(), a.end(), 0);
    0, 1, 2, 3, 4, ..., a.size() - 1
sort(a.begin(), a.end(), cmp2);
```

cmp возвращает true если первый элемент должен идти

раньше второго

```
stable_sort(a.begin(), a.end(), cmp);
```



```
auto cmp=[](int a, int b){ return a<b;};  
sort(a.begin(),a.end(),cmp);
```

```
auto cmp=[](int a, int b){ return a<b;};  
sort(a.begin(),a.end(),cmp);  
  
sort(a.begin(),a.end(),[](int a, int b){  
    return a < b;});
```

```
auto cmp=[](int a, int b){ return a<b;};  
sort(a.begin(),a.end(),cmp);
```

```
sort(a.begin(),a.end(),[](int a, int b){  
    return a < b;});
```

```
vector<int> x = {25,7,13};  
vector<int> a(3);  
iota(a.begin(),a.end(),1);  
sort(a.begin(),a.end(), [&](int i, int j){  
    return x[i] < x[j];});
```

[&] — видимость внутри всех переменных, которые видны в данном месте

```
vector<int> a(n);  
int b[n];  
for (int x:a){  
    cout << x;  
}
```



```
vector<int> a(n);  
int b[n];  
for (int x:a){  
    cout << x;  
}  
  
for (int& x:b){  
    cin >> x;  
}
```

```
vector<int> a(n);  
int b[n];  
for (int x:a){  
    cout << x;  
}  
  
for (int& x:b){  
    cin >> x;  
}  
  
for (auto x:b){  
    sum += x;  
}
```

```
vector<int> a(n);  
int b[n];  
for (int x:a){  
    cout << x;  
}  
  
for (int& x:b){  
    cin >> x;  
}  
  
for (auto x:b){  
    sum += x;  
}  
  
for (auto& x:a){  
    x = 1;  
}
```