



**KEEP
CALM
AND
LOVE
VECTORS**

std::vector

- *#include <vector>*
- Динамически расширяющийся массив
- Осуществляет проверку выхода за границы

std::vector [usage]

```
std::vector<int> v1; // empty vector
```

```
std::size_t n;  
std::cin >> n;
```

```
std::vector<double> v2(n); // vector with n zero elements
```

```
std::vector<char> v3(10, 'f'); // vector with 10 'f' symbols
```

std::vector [usage:2]

```
std::size_t n;  
std::cin >> n;  
  
std::vector<int> v(n);  
  
// Fill vector from console  
for(std::size_t i = 0; i < v.size(); ++i)  
{  
    std::cin >> v.at(i);  
}  
  
// Print reversed  
for(std::size_t i = v.size(); i > 0; --i)  
{  
    std::cout << v.at(i - 1) << ' '  
}  
  
std::cout << std::endl;
```

std::vector [usage:3]



```
std::vector<std::string> v;
```

```
v.push_back("The");  
v.push_back("bomb");  
v.push_back("has");  
v.push_back("been");  
v.push_back("planted");
```

```
for(std::size_t i = 0; i < v.size(); ++i)  
{  
    std::cout << v.at(i) << ' ' << '\n';  
}
```

```
std::cout << std::endl;
```

std::vector [usage:3.1] (C++11)

```
std::vector<std::string> v;

v.push_back("The");
v.push_back("bomb");
v.push_back("has");
v.push_back("been");
v.push_back("planted");

for(const auto& elem: v)
{
    std::cout << elem << ' ';
}

std::cout << std::endl;
```

std::vector methods

- **at(n)** – доступ элемент по индексу
 - Контролирует выход за границы массива
- **size()** – возвращает размер массива
- **push_back(x)** – добавляет элемент в конец
- **resize(n)** – изменяет размер массива
- ... [cppreference.com]

Problems

- [Timus #2023](#) (Дональд-почтальон)
- [Timus #1079](#) (Максимум)
- [Timus #1119](#) (Метро)