

```
video-Vector.cpp x
1  #include<cstdio>
2  #include<iostream>
3  #include<vector>
4  #include<algorithm>
5
6  using namespace std;
7
8  void showVec(vector<int> v2, string s = "")
9  {
10     cout<<s<< endl;
11     for (int i = 0; i < (int)v2.size(); ++i)
12         cout<<v2[i]<<" ";
13     cout<<"\n";
14 }
15
16 void demoBasicVecOperations()
17 {
18     vector<int> vecEmp;
19     showVec(vecEmp, "vecEmp: ");
20     getchar();
21
22     vecEmp.push_back(5);
23     showVec(vecEmp, "After inserting 5 in vecEmp: ");
24     getchar();
25
```

```

26     vector<int> v(10); // Initialize 10 elements
27     showVec(v, "v: ");
28     getchar();
29
30
31     vector<int> v1(10, 7); // Initialize 10 elements with value 7
32     showVec(v1, "v1: ");
33     getchar();
34     //we can even declare a vector of string or structures,
35     // which you have learned in C (Structured Programming) course.
36
37     // What about vector from array
38     const int lenght = 7;
39     int arr[lenght] = {7, 2, 5, 8, 2, 3, 9};
40     vector<int> v2(arr, arr+lenght);
41
42     showVec(v2, "v2: ");
43     getchar();
44
45     cout<<"Front: "<<v2.front()<<" last element: "<<v2.back()<<"\n";
46     getchar();
47
48     swap(v2.front(), v2.back()); // back is by reference
49     showVec(v2, "swap: ");
50     getchar();
51

```

```

51
52 // What about iterating on vector, Either as in
53 //array or using iterators 0 based indexing
54
55 // watch out: v.size() return unsigned integer.
56 // if you Return v.size()-1 may cause problems.
57 // So, Cast it to int
58 for (int i = 0; i < (int)v2.size(); ++i)
59     cout<<v2[i]<<" ";
60 cout<<"\n";
61 getchar();
62
63 // Another way: we will see later in details for
64 vector<int>::iterator it = v2.begin();
65 while(it != v2.end())
66 {
67     cout<<(*it)<<" "; // * works so similar
68     ++it; // Move to next time
69 }
70 cout<<"\n";
71 getchar();
72
73 // Vector from Vector
74 vector<int> v3(v1.begin(), v1.end());

```

```
75 showVec(v3, "v3 from v1: ");
76 getchar();
77
78 v.resize(15); // resize array from 10 to 15
79
80
81 // Sorting the vector
82 sort(v2.begin(), v2.end());
83 showVec(v2, "v2 sorted: ");
84 getchar();
85 // Sorting decreasing
86 sort(v2.begin(), v2.end(), greater<int>()); // let greater fir:
87 showVec(v2, "v2 sorted with : greater<int>() ");
88 getchar();
89
90 sort(v2.begin(), v2.end());
91 showVec(v2, "v2 sorted Again: ");
92 getchar();
93
94 //normally sorted in ascending and the pointers are changed
95 //end to begin so the it will create a descending order sorted
96 // vector instead of an ascending order sorted vector
97 sort(v2.rbegin(), v2.rend());
98 showVec(v2, "sorted reversed: ");
99 getchar();
```

```
100
101 // Creating 2D vector of boolean
102 vector< vector<bool> > vbs(10, vector<bool>(20, true));
103 vector<int> vd[10];
104
105 // ** Comparing 2 vectors, how???
106 // if(v1 < v2) // same <=, ==, >
107 //     cout<<"v1 Smaller than v2\n";
108 // else
109 //     cout<<"v2 Smaller than v1\n";
110
111 // Removing 2rd element
112 v2.erase(v2.begin()+1); // O(n)
113 showVec(v2, "erase 2nd: ");
114 getchar();
115
116 v2.insert(v2.begin()+3, 14); // O(n)
117 showVec(v2, "insert 14 in 4th location: ");
118 getchar();
119
120 v2.push_back(20);
121 showVec(v2, "add 20 in back: "); // O(1)
122 getchar();
123
124
```

```
125 // Check if element in array
126
127 if (find(v2.begin(), v2.end(), 7) != v2.end()) { // find is 0
128     cout<<"7 in v2\n";
129     cout << "7 is found in pos ";
130     cout << find(v2.begin(), v2.end(), 7)-v2.begin() << endl;
131 }
132
133 getchar();
134
135 if (find(v2.begin(), v2.end(), 100) == v2.end()) {
136     cout<<"100 is not in v2\n";
137 }
138 getchar();
139
140 // Removing value 7. If element not found, you can't call. Please
141 v2.erase( find(v2.begin(), v2.end(), 7) );
142 showVec(v2, "after erasing 7 using find: ");
143 getchar();
144
145 reverse(v2.begin(), v2.end() );
146 showVec(v2, "Reversed vector: ");
147 getchar();
148
149 random_shuffle(v2.begin(), v2.end());
150 showVec(v2, "Shuffled vector: ");
```

```
151     getchar();
152
153     v2.pop_back(); // remove last in O(1)
154     showVec(v2, "Last removed: ");
155     getchar();
156
157     v2.clear();
158     showVec(v2, "Cleared vector: ");
159     getchar();
160
161     // // inserts at the beginning
162     // v.emplace(v.begin(), 5);
163     // cout << "\nThe first element is: " << v[0];
164     //
165     // // Inserts 20 at the end
166     // v.emplace_back(20);
167     // int n = v.size();
168     // cout << "\nThe last element is:" << v[n - 1];
169     showVec(v1, "vector v1: ");
170     getchar();
171
172     // Swaps v1 and v2
173     v1.swap(v2);
174
175     showVec(v1, "\nAfter Swaps \nVector 1: ");
176     getchar();
```

```
177     showVec (v2, "Vector 2: ");
178     getchar ();
179
180     return;
181 }
182
183 int main()
184 {
185     demoBasicVecOperations ();
186     return 0;
187 }
188
189 /*
190 Reference:
191 1. Geeks for geeks
192 2. Mostafa Saad Ibrahim, FCU
193 */
194
```