

lab1.cpp

```
1  #include<iostream>
2  #include <unordered_set>
3  using namespace std;
4  void display(vector<int>vec) {
5      for (int i = 0; i < vec.size(); i++) {
6          cout << vec[i] << " ";
7      }
8      cout << endl;
9  }
10 void append(int x,vector<int>&vec){
11     cout << "Before Appending ";
12     for (auto t:vec){
13         cout << t << " ";
14     }
15     cout << endl;
16     vec.emplace_back(x);
17     cout << "After Appending ";
18     for (auto t:vec){
19         cout << t << " ";
20     }
21     cout << endl;
22 }
23 void insert(int index,int x,vector<int>&vec){
24     cout << "Before inserting ";
25     for (auto t:vec){
26         cout << t << " ";
27     }
28     cout << endl;
29     if (index <= vec.size())
30     {
31         vec.insert(vec.begin() + index, x);
32     }
33     else
34     {
35         cout << "Invalid index";
36     }
37     cout << "After inserting ";
38     for (auto t:vec){
39         cout << t << " ";
40     }
41     cout << endl;
42 }
43 void Delete(int x,vector<int>&vec){
44     cout << "Before deleting " <<x;
45     cout << endl;
46     display(vec);
47     int index = -1;
48     for (int i = 0; i < vec.size(); ++i)
49     {
50         if(vec[i] == x)
51             index = i;
52     }
53     vec.erase (vec.begin()+index);
54     cout << "After deleting " <<x;
55     cout << endl;
56     display(vec);
57 }
```

```
58 void LinearSearch(int x,vector<int>&vec){
59     int index = -1;
60     for (int i = 0; i < vec.size(); ++i)
61     {
62         if(vec[i] == x)
63             index = i;
64     }
65     if(index ==-1){
66         cout << x << " Not found";
67     }
68     else
69     {
70         cout << "Found " << x <<" at index " << index;
71     }
72 }
73 void Get(int index,vector<int>&vec){
74     if(index > vec.size()){
75         cout << "index out of bound ";
76         return;
77     }
78     for (int i = 0; i < vec.size(); ++i)
79     {
80         if(i == index)
81             cout << "Element " << vec[index];
82     }
83 }
84 void Set(int index,int x,vector<int>&vec){
85     cout << "Before inserting ";
86     cout << endl;
87     display(vec);
88     if (index <= vec.size())
89     {
90         vec.insert(vec.begin() + index, x);
91     }
92     else
93     {
94         cout << "Invalid index";
95     }
96     cout << "After inserting ";
97     cout << endl;
98     display(vec);
99 }
100 void Max(vector<int>&vec){
101     cout << "Array ";
102     cout << endl;
103     display(vec);
104     int max = INT_MIN;
105     for(auto it: vec){
106         if(it > max)
107             max = it;
108     }
109     cout << "Max " << max;
110 }
111 void Min(vector<int>&vec){
112     cout << "Array ";
113     cout << endl;
114     display(vec);
115     int min = INT_MAX;
116     for(auto it: vec){
117         if(it < min)
```

```
118         min = it;
119     }
120     cout << "Min " << min;
121 }
122 void Reverse(vector<int>&vec){
123     cout << "Array before reversing ";
124     cout << endl;
125     display(vec);
126     int l = 0;
127     int h = vec.size()-1;
128     while(l<h){
129         swap(vec[l],vec[h]);
130         l++;
131         h--;
132     }
133     cout << "Array after reversing ";
134     cout << endl;
135     display(vec);
136 }
137 void RightShift(vector<int>& vec, int k) {
138     cout << "Array before Right Shift ";
139     cout << endl;
140     display(vec);
141     for(int i=0;i<k;i++){
142         vec.pop_back();
143         vec.insert(vec.begin(),0);
144     }
145     cout << "Array after Right Shift ";
146     cout << endl;
147     display(vec);
148 }
149 void _reverseArray(vector<int>& nums,int s, int e){
150     while(s<e){
151         swap(nums[s],nums[e]);
152         s++;
153         e--;
154     }
155 }
156 void Rotate(vector<int>& vec, int k) {
157     cout << "Array before rotate by "<<k;
158     cout << endl;
159     display(vec);
160     int length = vec.size();
161     k = k % length;
162     _reverseArray(vec,0,length-1);
163     _reverseArray(vec,0,k-1);
164     _reverseArray(vec,k,length-1);
165     cout << "Array After rotate ";
166     cout << endl;
167     display(vec);
168 }
169
170 int main()
171 {
172     cout << "[i] Display()";
173     cout << endl;
174     vector<int> myvecOne = {1, 2, 3, 4, 5, 7, 9, 11, 13};
175     display(myvecOne);
176     cout << endl;
177     cout << endl;
```

```
178
179     cout << "[ii] Add/Append(x)";
180     cout << endl;
181     vector<int> myvecTwo = {5, 4, 3, 2, 1};
182     append(10, myvecTwo);
183     cout << endl;
184     cout << endl;
185
186     cout << "[iii] Insert(index, x)";
187     cout << endl;
188     vector<int> myvecThree = {5, 4, 3, 2, 1};
189     insert(3, 99, myvecThree);
190     cout << endl;
191     cout << endl;
192
193     cout << "[iv] Delete(x)";
194     cout << endl;
195     vector<int> myvecFour = {5, 4, 3, 2, 1};
196     Delete(3, myvecFour);
197     cout << endl;
198     cout << endl;
199
200     cout << "[v] LinearSearch(x)";
201     cout << endl;
202     vector<int> myvecFive = {5, 4, 3, 2, 1};
203     LinearSearch(4, myvecFive);
204     cout << endl;
205     cout << endl;
206
207     cout << "[vi] Get(index); function to get value available on the given index";
208     cout << endl;
209     vector<int> myvecSix = {5, 4, 3, 2, 1};
210     Get(2, myvecSix);
211     cout << endl;
212     cout << endl;
213
214     cout << "[vii] Set(index, x); insert the value x at the given index";
215     cout << endl;
216     vector<int> myvecSeven = {5, 4, 3, 2, 1};
217     Set(3,30, myvecSeven);
218     cout << endl;
219     cout << endl;
220
221     cout << "[viii] Max()";
222     cout << endl;
223     vector<int> myvecEight = {5, 4, 3, 2, 100};
224     Max(myvecEight);
225     cout << endl;
226     cout << endl;
227
228     cout << "[ix] Min()";
229     cout << endl;
230     vector<int> myvecNine = {5, 4, 3, 2, 100};
231     Min(myvecNine);
232     cout << endl;
233     cout << endl;
234
235     cout << "[x] Reverse()";
236     cout << endl;
237     vector<int> myvecTen = {5, 4, 3, 2, 100};
```

```
238 |     Reverse(myvecTen);
239 |     cout << endl;
240 |     cout << endl;
241 |
242 |     cout << "[xi] Shift()";
243 |     cout << endl;
244 |     vector<int> myvecEleven = {5, 4, 3, 2, 100};
245 |     RightShift(myvecEleven,3);
246 |     cout << endl;
247 |     cout << endl;
248 |
249 |     cout << "[xii] Rotate()";
250 |     cout << endl;
251 |     vector<int> myvecTwelve = {5, 4, 3, 2, 100};
252 |     Rotate(myvecTwelve,2);
253 |     cout << endl;
254 |     cout << endl;
255 | }
```