

```
1  #include<iostream>
2  #include <vector>
3  #include <cstdlib>
4  #include <ctype.h>
5  #include <cstring>
6  #include <time.h>
7  #include <random>
8  using namespace std;
9
10 void BubbleSort(vector<int> vect, int n);
11 void SelectionSort(vector<int> vect, int n);
12 void InsertionSort(vector<int> vect, int n);
13 void domandaSort(vector<int> vect, int n);
14
15
16 int main()
17 {
18     random_device rd;
19     mt19937 gen(rd()); // inizializzo il generatore pseudo
20
21
22     vector<int> vect;
23     int n, i, j, temp, x = 0;
24     string risposta;
25
26
27     cout << "Quanti elementi vuoi mettere nel vettore?" << endl;
28     cin >> n;
29
30     cout << "Vuoi che i numeri siano generati casualmente? Digita 1      ↗
31         \nVuoi che immettere i numeri manualmente? Digita 2" << endl;
32     cin >> risposta;
33
34
35     if (risposta == "1") {
36
37         int r1, r2;
38
39         cout << "Range dei numeri casuali (inserisci valori interi) :      ↗
40             \nRange1: ";
41         cin >> r1;
42         cout << "Range2: ";
43         cin >> r2;
44
45         uniform_int_distribution<> dis(r1, r2); //creo i numeri
46
47         cout << "\nI numeri generati sono: ";
48
49         for (int z = 0; z < n; z++)
50         {
51             int random = dis(gen);
52             cout << " " << random;
```

```
52         vect.push_back(random);
53     }
54
55     int sort = 0;
56     cout << "\n\nQuale metodo vuoi usare?\n1.Bubble Sort      ↗
        \n2.Selection Sort\n3.Insertion Sort(per numeri piccoli)\n  ↗
        \nDigita il numero del metodo: ";
57     cin >> sort;
58
59     switch (sort)
60     {
61     case 1:
62         BubbleSort(vect, n);
63     case 2:
64         SelectionSort(vect, n);
65     case 3:
66         InsertionSort(vect, n);
67     default:
68         break;
69     }
70
71 }
72 else if (risposta == "2") {
73     cout << "Inserisci gli elementi:\n";
74     for (i = 0; i < n; i++) {
75         cin >> x;
76         vect.push_back(x);
77     }
78
79     int sort = 0;
80     cout << "\n\nQuale metodo vuoi usare?\n1.Bubble Sort      ↗
        \n2.Selection Sort\n3.Insertion Sort (per numeri piccoli)\n  ↗
        \nDigita il numero del metodo: ";
81     cin >> sort;
82
83     switch (sort)
84     {
85     case 1:
86         BubbleSort(vect, n);
87     case 2:
88         SelectionSort(vect, n);
89     case 3:
90         InsertionSort(vect, n);
91     default:
92         break;
93     }
94
95 }
96 else {
97     cout << "Risposta non valida.";
98 }
99
100 return 0;
```

```
101 }
102
103
104 void domandaSort(vector<int> vect, int n) {
105
106
107 }
108
109 void BubbleSort(vector<int> vect, int n){
110
111     int temp, j, i;
112     int contatore = 0;
113     //ordino gli elementi
114     for (j = 0; j < n - 1; j++)
115         for (i = 0; i < n - 1; i++)
116             if (vect[i] > vect[i + 1])
117                 {
118                     //swap
119                     temp = vect[i];
120                     vect[i] = vect[i + 1];
121                     vect[i + 1] = temp;
122                     contatore++;
123                 }
124
125     cout << "\n\nArray ordinato con bubble sort:";
126     for (i = 0; i < n; i++)
127         cout << " " << vect[i];
128     cout << "\n\nNumero di operazioni: " << contatore;
129 }
130
131 void SelectionSort(vector<int> vect, int n) {
132     int temp, j, i;
133     int contatore = 0;
134     //ordino gli elementi
135     for (i = 0; i < n - 1; i++)
136         for (int j = i + 1; j < n; j++)
137             {
138                 if (vect[j] < vect[i]) {
139                     temp = vect[j];
140                     vect[j] = vect[i];
141                     vect[i] = temp;
142                     contatore++;
143                 }
144             }
145     cout << "\n\nArray ordinato con selection sort:";
146     for (i = 0; i < n; i++)
147         cout << " " << vect[i];
148     cout << "\n\nNumero di operazioni: " << contatore;
149 }
150
151 void InsertionSort(vector<int> vect, int n) {
152     int temp, j, i;
153     int contatore = 0;
```

```
154
155     //ordino gli elementi
156
157     for (i = 1; i < n; i++) {
158
159         temp = vect[i]; // i numero dopo, j numero prima
160         j = i - 1;
161
162         while (j >= 0 || vect[j] > temp ) {
163             vect[j + 1] = vect[j];
164             j--;
165             contatore++;
166         }
167         vect[j + 1] = temp;
168
169     }
170
171     cout << "\n\nArray ordinato con insertion sort:";
172     for (i = 0; i < n; i++)
173         cout << " " << vect[i];
174     cout << "\n\nNumero di operazioni: " << contatore;
175 }
176
```