```
1: #include <iostream>
 2: using namespace std;
 4: class Distance {
 5: private:
 6:
        int feet;
 7:
        float inches:
 8:
 9: public:
        // Constructor to initialize feet and inches to zero
10:
        Distance() {
11:
12:
            feet = 0;
13:
            inches = 0;
14:
15:
16:
        // Function to display the distance
17:
        void showdist() {
            cout << feet << " feet, " << inches << " inches" << endl;</pre>
18:
19:
20:
21:
        // Function to get distance from the user
22:
        void getdist() {
23:
            cin >> feet;
24:
            cin >> inches;
25:
        }
26:
27:
        // Function to set distance with provided values
28:
        void getdist(int f, float i) {
29:
            feet = f;
30:
            inches = i;
31:
        }
32:
33:
        // Function to add two distances
34:
        void sumdist(Distance d1, Distance d2) {
            int x;
35:
            float y;
36:
37:
38:
            x = d1.feet + d2.feet;
39:
            y = d1.inches + d2.inches;
40:
41:
            if (y >= 12) {
42:
                X++;
43:
                y = y - 12;
44:
45:
            feet = x;
            inches = y;
46:
47:
        }
48:
        // Function to subtract two distances
49:
        void mindist(Distance d1, Distance d2) {
50:
```

```
51:
             Distance max;
 52:
             Distance min;
 53:
 54:
             if (d1.feet == d2.feet) {
 55:
                  if (d1.inches <= d2.inches) {</pre>
 56:
                      min = d1;
 57:
                      max = d2:
 58:
                  } else {
 59:
                      min = d2;
                      max = d1;
 60:
61:
              } else if (d1.feet <= d2.feet) {</pre>
62:
63:
                  min = d1;
 64:
                  max = d2;
65:
              } else {
 66:
                  min = d2;
 67:
                  max = d1;
 68:
              }
 69:
70:
             int x = max.feet - min.feet;
71:
             float y = max.inches - min.inches;
72:
73:
             if (y < 0) {
 74:
                  y = 12 + y;
75:
                  x--;
76:
              }
77:
 78:
             feet = x;
79:
              inches = y;
 80:
81:
82:
         // Function to swap two Distance objects
83:
         void swap(Distance arr[], int i, int j) {
 84:
             Distance temp = arr[i];
 85:
              arr[i] = arr[j];
 86:
              arr[j] = temp;
 87:
         }
 88:
 89:
         // Function to sort an array of Distance objects in ascending order
90:
         void selectionSort(Distance arr[], int n) {
              for (int i = 0; i < n - 1; i++) {
91:
92:
                  int min = i;
93:
                  for (int j = i + 1; j < n; j++) {
94:
                      if (arr[j].feet == arr[min].feet) {
95:
                           if (arr[j].inches < arr[min].inches) {</pre>
96:
                               min = j;
97:
98:
                      } else if (arr[j].feet < arr[min].feet) {</pre>
99:
                          min = j;
100:
                      }
```

```
101:
                  if (min != i) {
102:
103:
                      swap(arr, min, i);
104:
105:
              }
106:
107:
              cout << "Sorted array is: " << endl;</pre>
              for (int i = 0; i < n; i++) {</pre>
108:
109:
                  arr[i].showdist();
110:
111:
         }
112: };
113:
114: int main() {
115:
          int n;
116:
          cout << "Enter length of your distance array: ";</pre>
117:
          cin >> n;
118:
119:
         Distance arr[n], temparr;
120:
         cout << "Enter the feet and inches of each distance: " << endl;</pre>
121:
         for (int i = 0; i < n; i++) {
              cout << "Enter distance " << i + 1 << endl;</pre>
122:
123:
              arr[i].getdist();
124:
         }
125:
126:
         Distance sum, diff;
          sum.sumdist(arr[0], arr[1]);
127:
128:
          diff.mindist(arr[0], arr[1]);
129:
130:
         cout << "Distance obtained after adding first two distances is:" << endl;</pre>
          sum.showdist();
131:
132:
         cout << "Distance obtained after subtracting first two distances is:" << endl;</pre>
133:
134:
         diff.showdist();
135:
136:
         temparr.selectionSort(arr, n);
137:
138:
         return 0;
139: }
140:
```