

```
1 // Justin Dang Student ID: 1148267
2 /*
3 Sorts char arrays and int arrays
4
5 known to work with the hard coded data and may not work with different arrays and ↗
   different sizes
6
7 when sorting strings, null char will remain when sorting
8 */
9 #include <iostream>
10 using namespace std;
11
12 template<typename T>
13 void Sort(T a[], int arrSize) {
14     T temp;
15     if (a[0] >= 'A' && a[0] <= 'a') { // tests if the array is not ↗
        // an int array by checking if the first index is between a range of ascii ↗
        // values
16         for (int i = 0; i < arrSize - 1; i++) { // sorts char array
17             for (int x = 0; x < arrSize - 1; x++) {
18                 if (a[x] > a[x + 1]) {
19                     temp = a[x];
20                     a[x] = a[x + 1];
21                     a[x + 1] = temp;
22                 }
23             }
24         }
25     }
26     else {
27         for (int i = 0; i < arrSize - 1; i++) { // sorts int array
28             for (int x = 0; x < arrSize - 1; x++) {
29                 if (a[x] > a[x + 1]) {
30                     temp = a[x];
31                     a[x] = a[x + 1];
32                     a[x + 1] = temp;
33                 }
34             }
35         }
36     }
37 }
38 template<typename T>
39 void Print(T a[], int arrSize) {
40     for (int i = 0; i < arrSize; i++)
41         cout << a[i] << ' ';
42 }
43
44 int main()
45 {
46     const int arr1Size = 10, arr2Size = 7; // array sizes
47     int arr1[arr1Size] = {10, 4, 9, 1, 2, 5, 6, 8, 7, 3}; // int array
48     char arr2[arr2Size] = { 'J', 'u', 's', 't', 'i', 'n' }; // char array
49 }
```

```
50     cout << "After sorting:\n\narr1: ";
51     Print(arr1, arr1Size);
52     cout << "\n\narr2: ";
53     Print(arr2, arr2Size);
54
55     cout << "\n\n";
56
57     Sort(arr1, arr1Size);
58     Sort(arr2, arr2Size);
59
60     cout << "After sorting:\n\narr1: ";
61     Print(arr1, arr1Size);
62     cout << "\n\narr2: ";
63     Print(arr2, arr2Size);
64 }
65 /*//-----case 1:
66 After sorting:
67
68 arr1: 10 4 9 1 2 5 6 8 7 3
69
70 arr2: J u s t i n
71
72 After sorting:
73
74 arr1: 1 2 3 4 5 6 7 8 9 10
75
76 arr2:  J i n s t u
77 *//-----
78
```