Source File: `~/2336/20/lab20.cpp`
Input: under control of `main` function
Output: under control of `main` function
Value: 2

Write a function template whose prototype is given by

```
1 template<typename T>
2 void mySwap(T& first, T& second);
```

The function swaps the contents of `first` and `second`. Write a second function template that is **recursive** and whose prototype is given by

```
1 template<typename T>
2 void bubbleSort(T *array, int n);
```

`bubbleSort` sorts the elements in `[array, array + n)` into ascending order, meaning that if `i` and `j` are any two valid addresses in `[array, array + n)` such that `i` precedes `j`, then `*j` is not less than `*i`. The sort algorithm should incorporate two enhancements: the length of the array should be diminished by one on each recursive call and the algorithm should cease recursing if no swaps are made on a pass through the elements. A `main` function for testing your functions is shown in Figure 1. The expected output from executing this code is shown in Figure 2. To use the `Makefile` as distributed in class, add a target of `lab20main` to `targets1srcfile`.

A discussion of the bubble sort can be found in the 9th edition of the Gaddis book on pp. 476–481. The code shown is **iterative**.

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // function template prototypes
7  template<typename T>
8  void mySwap(T& first, T& second);
9  template<typename T>
10 void bubbleSort(T *array, int n);
11
12 #include "lab20.cpp"
13
14 template<typename T>
15 void printArray(const T *array, int count)
16 {
17   if (count > 0)
18   {
19     cout << *array << " ";
20     printArray(array + 1, count - 1);
21   }
22   else
23     cout << endl;
24 }
25
```

Figure 1. `/usr/local/2336/src/lab20main.C` (Part 1 of 2)

```
26 int main()
27 {
28   const int aCount = 5, bCount = 7, cCount = 5, dCount = 8;
29   int a[aCount] = {5, 4, 3, 2, 1};
30   double b[bCount] = {7.7, 6.6, 5.5, 4.4, 3.3, 2.2, 1.1};
31   char c[cCount] = {'H', 'E', 'L', 'L', 'O'};
32   string d[dCount] = {"Cadillac", "Oldsmobile", "Chevrolet",
33                       "Toyota", "Lexus", "Dodge", "GMC", "BMW"};
34
35   cout << "Array a contains:" << endl;
36   printArray(a, aCount);      // integer print function template
37   bubbleSort(a, aCount);      // integer sort function template
38   printArray(a, aCount);      // integer print function template
39
40   cout << "Array b contains:" << endl;
41   printArray(b, bCount);      // double print function template
42   bubbleSort(b, bCount);      // double sort function template
43   printArray(b, bCount);      // double print function template
44
45   cout << "Array c contains:" << endl;
46   printArray(c, cCount);      // character print function template
47   bubbleSort(c, cCount);      // character sort function template
48   printArray(c, cCount);      // character print function template
49
50   cout << "Array d contains:" << endl;
51   printArray(d, dCount);      // string print function template
52   bubbleSort(d, dCount);      // string sort function template
53   printArray(d, dCount);      // string print function template
54
55   return 0;
56 }
```

Figure 1. `/usr/local/2336/src/lab20main.C` (Part 2 of 2)

```
newuser@csunix ~> cd 2336
newuser@csunix ~/2336> ./getlab.ksh 20
  * Checking to see if a folder exists for Lab 20. . .No
  * Creating a folder for Lab 20
  * Checking to see if Lab 20 has sample input and output files. . .Yes
  * Copying input and output files for Lab 20
    from folder /usr/local/2336/data/20 to folder ./20
  * Checking to see if /usr/local/2336/src/lab20main.C exists. . .Yes
  * Copying file /usr/local/2336/src/lab20main.C to folder ./20
  * Checking to see if /usr/local/2336/include/lab20.h exists. . .No
  * Copying file /usr/local/2336/src/Makefile to folder ./20
  * Adding a target of lab20main to targets1srcfile
  * Touching file ./20/lab20.cpp
  * Edit file ./20/lab20.cpp in Notepad++
newuser@csunix ~/2336> cd 20
newuser@csunix ~/2336/20> ls
01.dat       01.out       Makefile     lab20.cpp    lab20main.C
newuser@csunix ~/2336/20> make lab20main
g++ -g -Wall -std=c++11 -c lab20main.C -I/usr/local/2336/include -I.
g++ -o lab20main lab20main.o -L/usr/local/2336/lib -lm -lbits
newuser@csunix ~/2336/20> ./lab20main
Array a contains:
5 4 3 2 1
1 2 3 4 5
Array b contains:
7.7 6.6 5.5 4.4 3.3 2.2 1.1
1.1 2.2 3.3 4.4 5.5 6.6 7.7
Array c contains:
H E L L O
E H L L O
Array d contains:
Cadillac Oldsmobile Chevrolet Toyota Lexus Dodge GMC BMW
BMW Cadillac Chevrolet Dodge GMC Lexus Oldsmobile Toyota
newuser@csunix ~/2336/20> ./lab20main > my.out
newuser@csunix ~/2336/20> diff 01.out my.out
newuser@csunix ~/2336/20>
```

Figure 2. Commands to Compile, Link, & Run Lab 20