**A Dynamic Array**

Write a C++ program that does the following:

1. Prompt the user for the number of values to be used, the maximum possible (integer) value, and the minimum possible (integer) value.
2. Dynamically allocate an array exactly large enough to hold that many integer values.
3. Fill the array with randomly generated values between the user-specified minimum and maximum values, inclusive.
4. Sort the array. You may use either bubble sort or selection sort.
5. Print to the screen:
   1. all the values, in order
   2. the smallest value in the array
   3. the largest value in the array
   4. the median value in the array

A run of your program should produce output that looks exactly like this:

Enter the number of values: 10

Enter the minimum value: -5

Enter the maximum value: 20

The values in order: [-4, -4, 1, 1, 7, 7, 8, 8, 18, 18]

The smallest value in the array is -4

The largest value in the array is 18

The median value is 7

or like this:

Enter the number of values: 10

Enter the minimum value: -5

Enter the maximum value: 15

The values in order: [-5, -3, -1, -1, -1, 2, 12, 13, 13, 15]

The smallest value in the array is -5

The largest value in the array is 15

The median value is 0.5

or this:

Enter the number of values: 9

Enter the minimum value: 5

Enter the maximum value: 12

The values in order: [5, 5, 5, 6, 7, 8, 8, 9, 9]

The smallest value in the array is 5

The largest value in the array is 9

The median value is 7

At a minimum you must implement functions to fill the array with random values, to sort the array, and to compute the median of the values in the array. You may define other functions if you wish.

Be sure you correctly de-allocate the dynamically created array before the program ends.

Remember that *every* variable that holds a size or an index (i.e., a position) must be declared as type size\_t.

Remember that to generate a random integer between min and max inclusive, we use: rand() % (max - min + 1) + min. Also remember that you must seed the random number generator exactly once in the program.

For an odd number of values, the median is the middle value. For an even number of values, the median is the average of the two middle values.