Arrays: Drills

1) Declare an array of size 10 of whichever data type you'd like. Initialize the array in your code (that is, don't let the user enter the data from the console, hard-code the values in). Print the array to the console.
2) Declare a double array of size 10. Using an appropriate loop, have the user enter all the values of the array. Print the array to the console.

3) Assume that we have an array:

```
int AR[10] = {0};
for(int i = 0; i < 10; i++)
{
    cin >> AR[i];
}
```

Output the maximum and minimum of the array

4) Create an array of size 10 of type double. Have the user enter the values (or hard-code them in, it is your choice). Find and output the average.

5) Create an integer array of size 10, and enter the values. Determine if the array is sorted ascending (ie: the smallest number is first, highest is last). Remember, the array {1,1,1,1,1,1,1,1,1,1} is, in fact, sorted ascending.

6) Create an integer array of size 10. Have the user enter values. Do not allow the user to enter a value already in the array. Stop entering values when there is no more space in the array. Print the array to the screen. For instance, if the user enters the number 2 twice, only the first number should be stored in the array, and the second should be ignored.

7) Create an integer array of size 10, and have the user enter the values from the console. Using as many new arrays as you need, print out the UNIQUE elements of the first array. Example:

```
int AR[10] = {1,1,1,2,3,4,5,5,6,7};
...code...
I want to see: 1, 2, 3, 4, 5, 6, 7
```

8) WARNING: SUPER HARD PROBLEM:
Create an integer array of size 5. Have the user enter values. When entering a new value, make sure to put it in the array so that the array is sorted, ascending. Print the sorted array.

Example: If the user enters 3, 4, 1, 5, 2
I want the array to store the numbers in the order
1, 2, 3, 4, 5