1. Create an integer array that will hold 100 ints. Fill this array with random number between 0 and 100. Print the contents of the array in 20 rows of 5 columns.

2. Write a program to find and print the minimum, maximum, and average value of the array created in 1 above.

3. Write a program to create an array of 100 ints which is called reverse. Use the array created in 1 above and fill reverse with the same information in reverse order. Print the first array, wait for the user to push a key, and print the second array. Both arrays should be printed in 20 rows of 5 columns.

4. Write a main program to create the following array:
   
   ```
   int a[] = {0, 1, 2, 3, 4, 5};
   ```

   Write a function called Shift which will shift all of the values in a array by n places where n is an argument. Your function prototype will look like the following:

   ```
   void Shift(int a[], int length, int places, bool direction);
   ```

   where a[] is the array to be operated on, length is the array size, places is the number of places to shift, and direction is a bool which indicates the direction of the shift (right or left). For example, if you call the function with the following statement:

   ```
   Shift(a, 6, 2, false);
   ```

   The values in a will be shifted to the left by two places so that the shifted array will have the following values: 2, 3, 4, 5, 0, 0. Note that zeros should be shifted in from the right on a left shift or from the left on a right shift.

5. Write a function called SwapElements which accepts an array and two ints as arguments. The function swaps the two elements whose index is given by the two ints. For example if we define an array as:

   ```
   int a[] = {0, 1, 2, 3, 4, 5};
   ```

   The following call would swap elements 2 and 4:

   ```
   SwapElements(a, 2, 4);
   ```

   The values in the array after the call would be: 0, 1, 4, 3, 2, 5.

6. Suppose we want to write a function which will sort an array. One sorting algorithm is called the select sort. Here is an example of how a select sort works: Consider an array that has just 4 elements defined by

   ```
   int a[4] = {4, 0, 3, 2};
   ```

   The select sort first finds the minimum value of the four elements which in this case is 0. It swaps this 0 with the first array element so that the array becomes: 0, 4, 3, 2. It then finds the minimum of all of the elements except for the first and swaps this minimum with the second element. The new array becomes 0, 2, 3, 4. In the next iteration, the algorithm finds the minimum of all of the elements except for the first two and swaps it with the third position. This continues until the array is completely sorted.

   Use the SwapElements function in 5 above to write a SelectSort function. The prototype is:

   ```
   void SelectSort(int a[], int length);
   ```

   where length is the size of the array.