```
// This program uses a selection sort to arrange an array of integers in
// ascending order

// MICHAEL STEELE

#include <iostream>
using namespace std;

// function prototypes
void selectionSortArray(int[], int);
void displayArray(int[], int);

const int SIZE = 8;

int main()
{
	int values[SIZE] = { 23, 0, 45, -3, -78, 1, -1, 9 };

	cout << "The values before the selection sort is performed are:" << endl;
	displayArray(values, SIZE);

	selectionSortArray(values, SIZE);

	cout << "The values after the selection sort is performed are:" << endl;
	displayArray(values, SIZE);

	return 0;
}

//******************************************************************
//	displayArray
//
//  task:  to print the array
//  data in:  the array to be printed, the array size
//  data out: none
//
//******************************************************************

void displayArray(int array[], int elems)	// function heading
{
	// Displays array
	for (int count = 0; count < elems; count++)
		cout << array[count] << " ";
	cout << endl;
}

//******************************************************************
//	selectionSortArray
//
//  task:  to sort values of an array in ascending order
//  data in:  the array, the array size
//  data out: the sorted array
//
//******************************************************************

void selectionSortArray(int array[], int elems)
{
	int seek;		// array position currently being put in order
	int minCount;  // location of smallest value found
	int minValue;  // holds the smallest value found

	for (seek = 0; seek < (elems - 1); seek++)    // outer loop performs the swap
												// and
then increments seek
	{
		minCount = seek;
		minValue = array[seek];

		for (int index = seek + 1; index < elems; index++)
		{
			// inner loop searches through array
			// starting at array[seek] searching
			// for the smallest value. When the
			// value is found, the subscript is
			// stored in minCount. The value is
			// stored in minValue.

			if (array[index] > minValue)
			{
				minValue = array[index];
				minCount = index;



			}
```

```
                    }

                    // the following two statements exchange the value of the
                    // element currently needing the smallest value found in the
                    // pass(indicated by seek) with the smallest value found
                    // (located in minValue)

                    array[minCount] = array[seek];
                    array[seek] = minValue;
                    for(int i = 0; i < elems; i++)
                    {


          cout << array[i] << " ";
                    }
                    cout << "swap " << seek+1 << endl;

          }
}
```

```
The values before the selection sort is performed are:
23 0 45 -3 -78 1 -1 9
45 0 23 -3 -78 1 -1 9 swap 1
45 23 0 -3 -78 1 -1 9 swap 2
45 23 9 -3 -78 1 -1 0 swap 3
45 23 9 1 -78 -3 -1 0 swap 4
45 23 9 1 0 -3 -1 -78 swap 5
45 23 9 1 0 -1 -3 -78 swap 6
45 23 9 1 0 -1 -3 -78 swap 7
The values after the selection sort is performed are:
45 23 9 1 0 -1 -3 -78

Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```