```cpp
// This program demonstrates a Binary Search

// Michael Steele

#include <iostream>
using namespace std;

int binarySearch(int[], int, int);        // function prototype
const int SIZE = 16;

int main()
{
        int found, value;

        int array[] = { 34, 19, 19, 18, 17, 13, 12, 12, 12, 11, 9, 5, 3, 2, 2, 0 };
        // array to be searched

        cout << "Enter an integer to search for:" << endl;
        cin >> value;

        found = binarySearch(array, SIZE, value);
        // function call to perform the binary search
        // on array looking for an occurrence of value

        if (found == -1)
                cout << "The value " << value << " is not in the list" << endl;
        else
                cout << "The value " << value << " is in position number "
                    << found + 1 << " of the list" << endl;

        return 0;
}

//********************************************************************
//        binarySearch
//
//  task:      This searches an array for a particular value
//  data in:         List of values in an orderd array, the number of
//                   elements in the array, and the value searched for
//                   in the array
//  data returned: Position in the array of the value or -1 if value
//                   not found
//
```

```
//********************************************************************

int binarySearch(int array[], int numElems, int value)       // function heading
{
        int first = 0;                               // First element of list
        int last = numElems - 1;        // last element of the list
        int middle;                                          // variable containing the current
                                                                     // middle value of the list

        while (first <= last)
        {
                middle = first + (last - first) / 2;

                if (array[middle] == value)
                        return middle;          // if value is in the middle, we are done

                else if (array[middle]<value)
                        last = middle - 1;        // toss out the second remaining half of

                else
                        first = middle + 1;       // toss out the first remaining half of
                                                                     // the array and search the second
        }

        return -1;       // indicates that value is not in the array
}
```

```
Enter an integer to search for:
2
The value 2 is in position number 14 of the list

Process returned 0 (0x0)   execution time : 5.225 s
Press any key to continue.
```