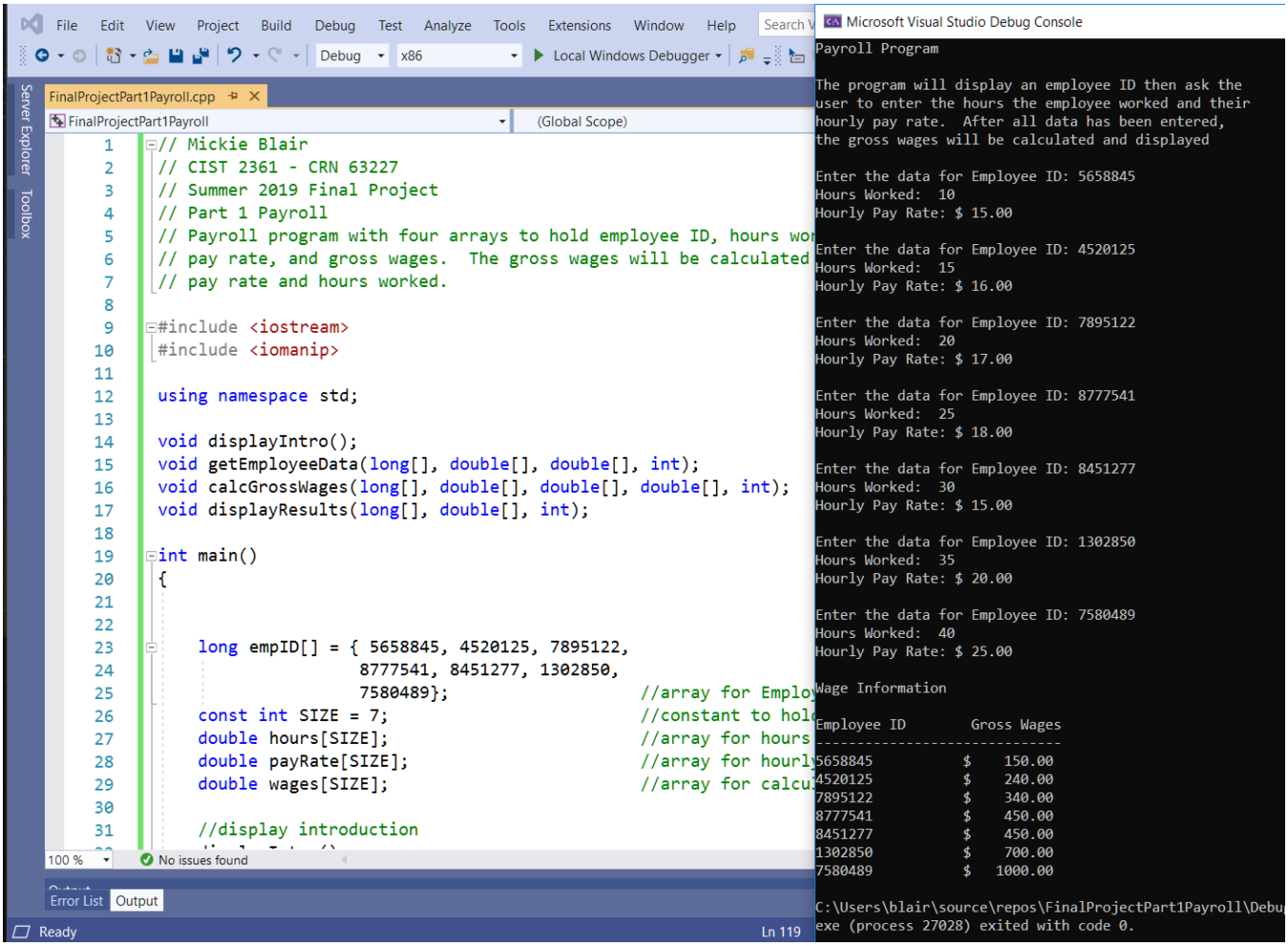
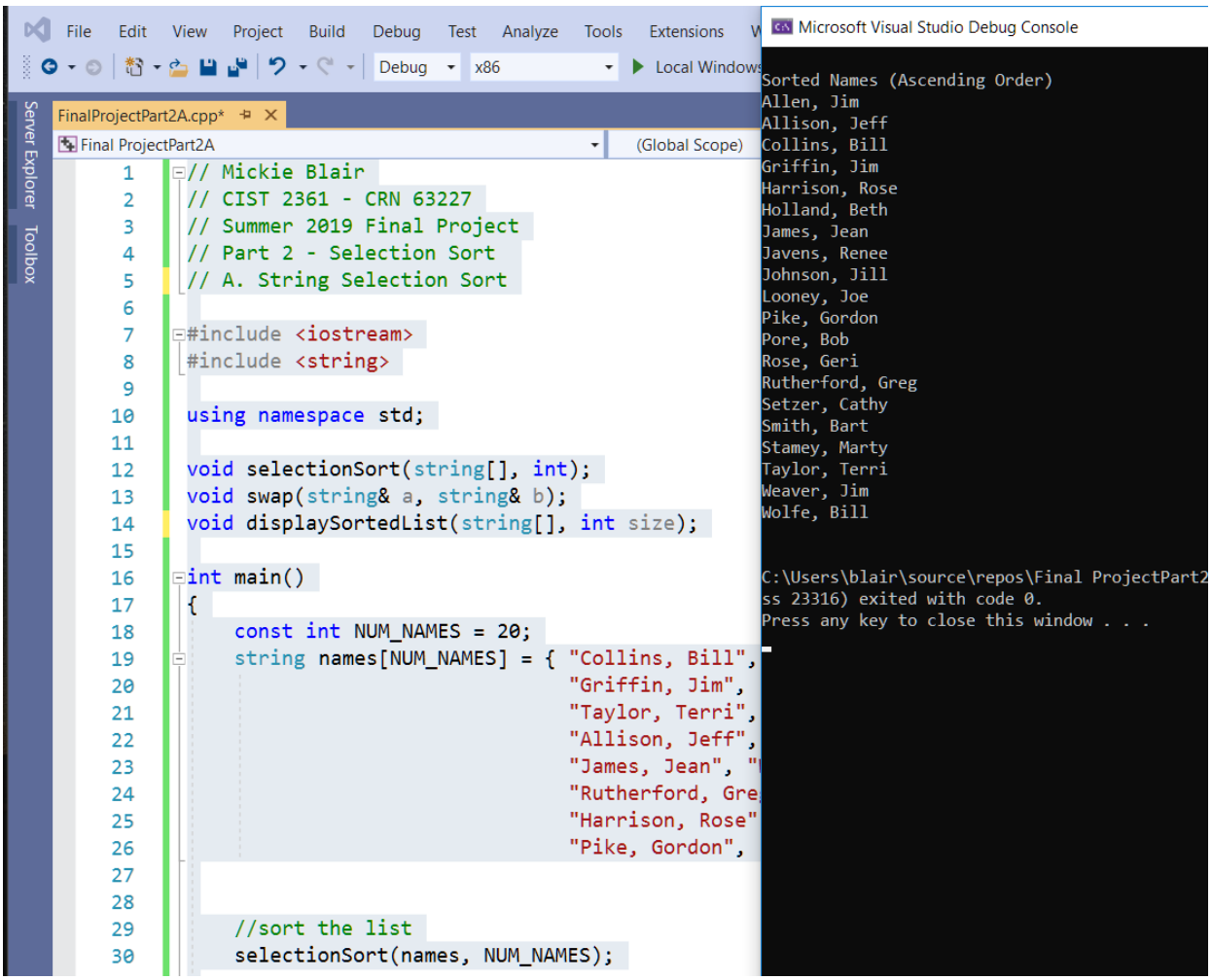
```
// Mickie Blair
// CIST 2361 - CRN 63227
// Summer 2019 Final Project
// Part 1 Payroll
// Payroll program with four arrays to hold employee ID, hours worked
// pay rate, and gross wages.  The gross wages will be calculated using
// pay rate and hours worked.

#include <iostream>
#include <iomanip>

using namespace std;

void displayIntro();
void getEmployeeData(long[], double[], double[], int);
void calcGrossWages(long[], double[], double[], double[], int);
void displayResults(long[], double[], int);

int main()
{


	long empID[] = { 5658845, 4520125, 7895122,
			 8777541, 8451277, 1302850,
			 7580489};                          //array for Employee ID
	const int SIZE = 7;                         //constant to hold size of arrays
	double hours[SIZE];                         //array for hours worked
	double payRate[SIZE];                       //array for hourly payrate
	double wages[SIZE];                         //array for calculated gross wages

	//display introduction
	displayIntro();

	//get the data
	getEmployeeData(empID, hours, payRate, SIZE);

	//calculate gross wages
	calcGrossWages(empID, hours, payRate, wages, SIZE);

	//display results
	displayResults(empID, wages, SIZE);


}

//display introduction
void displayIntro()
{
	cout << "Payroll Program\n" << endl;
	cout << "The program will display an employee ID then ask the " << endl;
	cout << "user to enter the hours the employee worked and their " << endl;
	cout << "hourly pay rate.  After all data has been entered," << endl;
	cout << "the gross wages will be calculated and displayed" << endl;
}

//get employee data from user
void getEmployeeData(long idNumber[], double hoursWorked[], double hourlyPay[],
                                        int arraySize)
{
	//step through employee ID array and ask user for the hours worked
	// and the hourly pay Rate.
```

```
	for (int index = 0; index < arraySize; index++)
	{
		cout << endl;
		cout << "Enter the data for Employee ID: " << idNumber[index];
		cout << "\nHours Worked:  ";
		cin >> hoursWorked[index];

		//validation loop
		while (hoursWorked[index] < 0.0)
		{
			cout << "The hours worked must be 0 or greater." << endl;
			cout << "\nHours Worked:  ";
			cin >> hoursWorked[index];
		}

		cout << "Hourly Pay Rate: $ ";
		cin >> hourlyPay[index];

		//validation loop
		while (hourlyPay[index] < 15.00)
		{
			cout << "The hourly pay must be $15.00 or greater." << endl;
			cout << "Hourly Pay Rate: $ ";
			cin >> hourlyPay[index];
		}
	}
}

//calculate gross wages
void calcGrossWages(long idNumber[], double hoursWorked[], double hourlyPay[],
	double grossWages[], int arraySize)
{
	for (int index = 0; index < arraySize; index++)
	{
		grossWages[index]= hoursWorked[index] * hourlyPay[index];
	}
}

//display results
void displayResults(long idNumber[], double grossWages[], int arraySize)
{
	cout << setprecision(2) << fixed;

	cout << "\nWage Information\n" << endl;

	cout << setw(15) << left << "Employee ID";
	cout<< setw(15) << right << "Gross Wages" << endl;
	cout << "------------------------------\n";

	for (int index = 0; index < arraySize; index++)
	{
		cout << setw(15) << left << idNumber[index];
		cout << setw(4) << right << "$";
		cout << setw(10) << right << grossWages[index] << endl;
	}
}
```


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Debug x86 Local Windows Debugger
FinalProjectPart1Payroll.cpp
FinalProjectPart1Payroll (Global Scope)
// Mickie Blair
// CIST 2361 - CRN 63227
// Summer 2019 Final Project
// Part 1 Payroll
// Payroll program with four arrays to hold employee ID, hours wo
// pay rate, and gross wages. The gross wages will be calculated
// pay rate and hours worked.
#include <iostream>
#include <iomanip>
using namespace std;
void displayIntro();
void getEmployeeData(long[], double[], double[], int);
void calcGrossWages(long[], double[], double[], double[], int);
void displayResults(long[], double[], int);
int main()
{
long empID[] = { 5658845, 4520125, 7895122,
8777541, 8451277, 1302850,
7580489};
const int SIZE = 7;
double hours[SIZE];
double payRate[SIZE];
double wages[SIZE];
//display introduction
100 %
No issues found
Error List
Output
Ready
Ln 119
Microsoft Visual Studio Debug Console
Payroll Program
The program will display an employee ID then ask the
user to enter the hours the employee worked and their
hourly pay rate. After all data has been entered,
the gross wages will be calculated and displayed
Enter the data for Employee ID: 5658845
Hours Worked: 10
Hourly Pay Rate: $ 15.00
Enter the data for Employee ID: 4520125
Hours Worked: 15
Hourly Pay Rate: $ 16.00
Enter the data for Employee ID: 7895122
Hours Worked: 20
Hourly Pay Rate: $ 17.00
Enter the data for Employee ID: 8777541
Hours Worked: 25
Hourly Pay Rate: $ 18.00
Enter the data for Employee ID: 8451277
Hours Worked: 30
Hourly Pay Rate: $ 15.00
Enter the data for Employee ID: 1302850
Hours Worked: 35
Hourly Pay Rate: $ 20.00
Enter the data for Employee ID: 7580489
Hours Worked: 40
Hourly Pay Rate: $ 25.00
Wage Information
Employee ID Gross Wages
5658845 $ 150.00
4520125 $ 240.00
7895122 $ 340.00
8777541 $ 450.00
8451277 $ 450.00
1302850 $ 700.00
7580489 $ 1000.00
C:\Users\blair\source\repos\FinalProjectPart1Payroll\Debu
exe (process 27028) exited with code 0.

```cpp
// Mickie Blair
// CIST 2361 - CRN 63227
// Summer 2019 Final Project
// Part 2 - Selection Sort
// A. String Selection Sort

#include <iostream>
#include <string>

using namespace std;

void selectionSort(string[], int);
void swap(string& a, string& b);
void displaySortedList(string[], int size);

int main()
{
	const int NUM_NAMES = 20;
	string names[NUM_NAMES] = { "Collins, Bill", "Smith, Bart", "Allen, Jim",
	                            "Griffin, Jim", "Stamey, Marty", "Rose, Geri",
	                            "Taylor, Terri", "Johnson, Jill",
	                            "Allison, Jeff", "Looney, Joe", "Wolfe, Bill",
	                            "James, Jean", "Weaver, Jim", "Pore, Bob",
	                            "Rutherford, Greg", "Javens, Renee",
	                            "Harrison, Rose", "Setzer, Cathy",
	                            "Pike, Gordon", "Holland, Beth" };


	//sort the list
	selectionSort(names, NUM_NAMES);

	//display the sorted list
	displaySortedList(names, NUM_NAMES);


	return 0;
}

//selection sort function

void selectionSort(string namesArray[], int size)
{
	int startScan;
	int minIndex;
	string minValue;

	for (startScan = 0; startScan < (size - 1); startScan++)
	{
		minIndex = startScan;
		minValue = namesArray[startScan];

		for (int index = startScan + 1; index < size; index++)
		{
			if (namesArray[index] < minValue)
			{
				minValue = namesArray[index];
				minIndex = index;
			}
		}

		swap(namesArray[minIndex], namesArray[startScan]);
	}
```

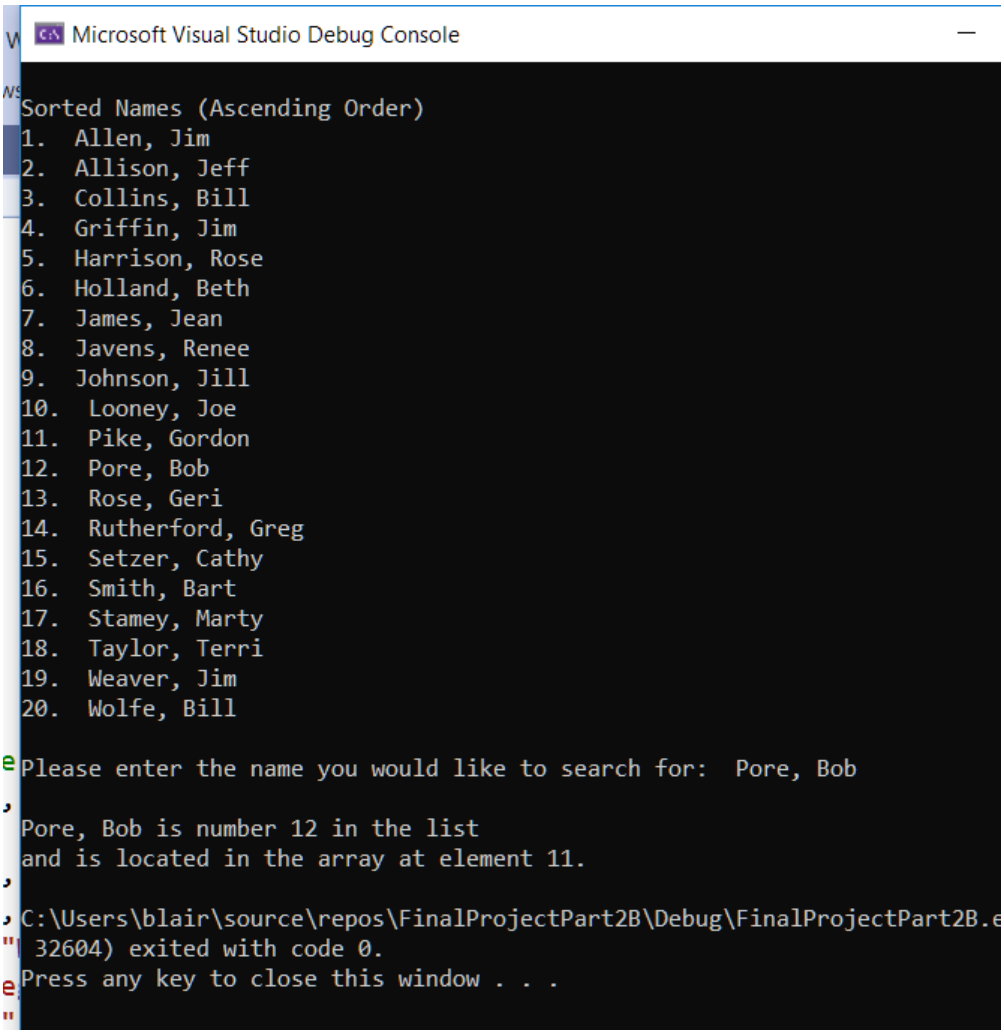
```
}


//swap function
void swap(string& a, string& b)
{
	string temp = a;
	a = b;
	b = temp;
}

//display the sorted list
void displaySortedList(string namesArray[], int size)
{
	//display the sorted array
	cout << "\nSorted Names (Ascending Order)\n";

	//for loop for displaying
	for (int count = 0; count < size; count++)
	{
		cout << namesArray[count] << endl;
	}

	cout << endl;
}
```

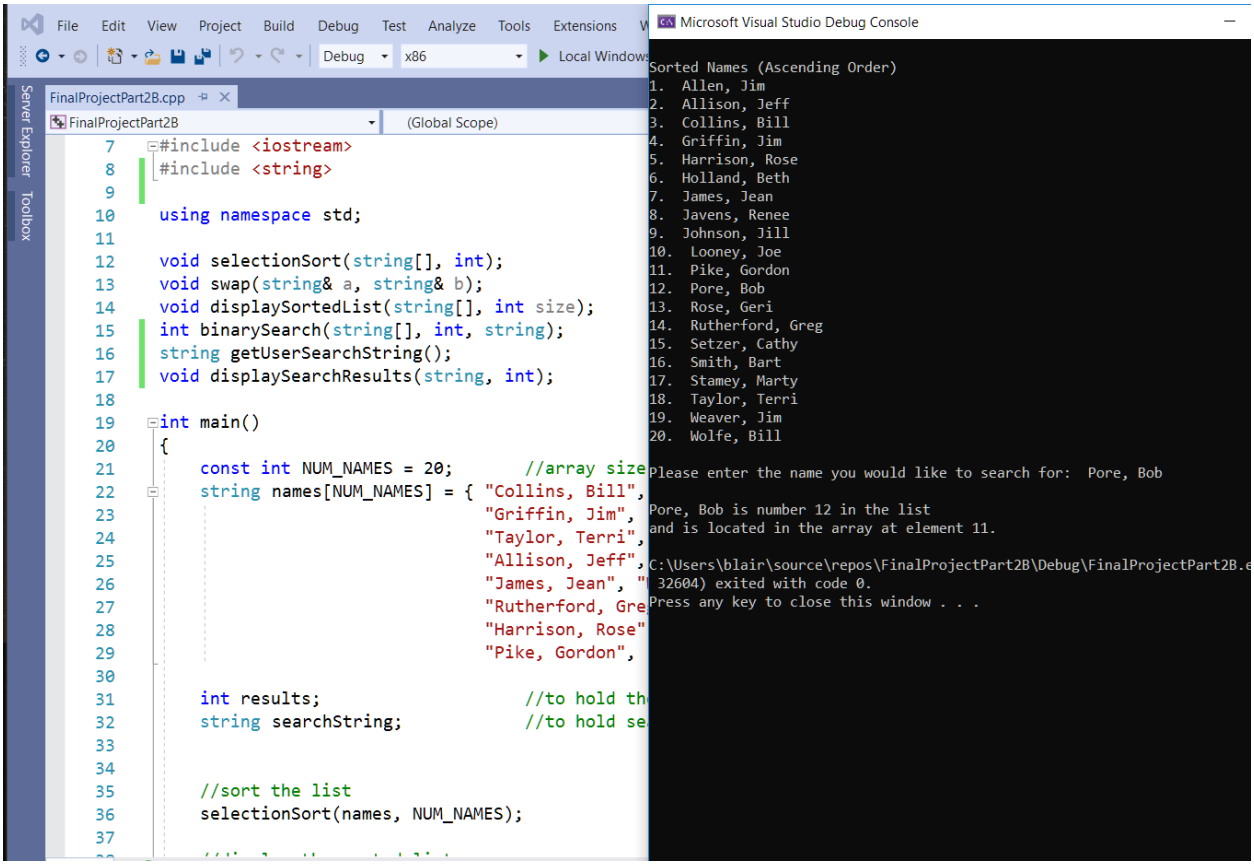


```
// Mickie Blair
// CIST 2361 - CRN 63227
// Summer 2019 Final Project
// Part 2 - Selection Sort
// B. Binary String Search

#include <iostream>
#include <string>

using namespace std;

void selectionSort(string[], int);
void swap(string& a, string& b);
void displaySortedList(string[], int size);
int binarySearch(string[], int, string);
string getUserSearchString();
void displaySearchResults(string, int);

int main()
{
	const int NUM_NAMES = 20;		//array size
	string names[NUM_NAMES] = { "Collins, Bill", "Smith, Bart", "Allen, Jim",
				    "Griffin, Jim", "Stamey, Marty", "Rose, Geri",
				    "Taylor, Terri", "Johnson, Jill",
				    "Allison, Jeff", "Looney, Joe", "Wolfe, Bill",
				    "James, Jean", "Weaver, Jim", "Pore, Bob",
				    "Rutherford, Greg", "Javens, Renee",
				    "Harrison, Rose", "Setzer, Cathy",
				    "Pike, Gordon", "Holland, Beth" }; //array of names

	int results;			//to hold the search results
	string searchString;		//to hold search value


	//sort the list
	selectionSort(names, NUM_NAMES);

	//display the sorted list
	displaySortedList(names, NUM_NAMES);

	//get user search value
	searchString = getUserSearchString();


	//search the array
	results = binarySearch(names, NUM_NAMES, searchString);


	//display the search result
	displaySearchResults(searchString, results);

	return 0;
}

//selection sort function

void selectionSort(string namesArray[], int size)
{
	int startScan;
	int minIndex;
	string minValue;
```

```
	for (startScan = 0; startScan < (size - 1); startScan++)
	{
		minIndex = startScan;
		minValue = namesArray[startScan];

		for (int index = startScan + 1; index < size; index++)
		{
			if (namesArray[index] < minValue)
			{
				minValue = namesArray[index];
				minIndex = index;
			}
		}

		swap(namesArray[minIndex], namesArray[startScan]);
	}
}


//swap function
void swap(string& a, string& b)
{
	string temp = a;
	a = b;
	b = temp;
}

//display the sorted list
void displaySortedList(string namesArray[], int size)
{
	//display the sorted array
	cout << "\nSorted Names (Ascending Order)\n";

	//for loop for displaying
	for (int count = 0; count < size; count++)
	{
		cout << (count + 1)<< ".  " << namesArray[count] << endl;
	}

	cout << endl;
}

string getUserSearchString()
{
	string input;
	cout << "Please enter the name you would like to search for:  ";
	getline(cin, input);

	return input;
}

int binarySearch(string namesArray[], int numElems, string value)
{
	int first = 0,			// First array element
	    last = numElems - 1,	// Last array element
	    middle,			// Midpoint of search
	    position = - 1;		// Position of search value
	bool found = false;		// Flag
	while (!found && first <= last)
	{
		middle = (first + last) / 2;		// Calculate midpoint
		if (namesArray[middle].compare(value)==0)	// If value is found at mid
```

```
		{
			found = true;
			position = middle;
		}
		else if (namesArray[middle].compare(value) > 0) //If value is in lower half
			last = middle - 1;
		else
			first = middle + 1;		// If value is in upper half
	}
	return position;
}

//display search results
void displaySearchResults(string search, int searchResult)
{
	if (searchResult == -1)
		cout << search << " is not in list.\n";
	else
	{
		cout << endl;
		cout << search << " is number " << searchResult + 1 << " in the list\n";
		cout << "and is located in the array at element " << searchResult << ".\n";
	}
}
```




File Edit View Project Build Debug Test Analyze Tools Extensions
Debug x86 Local Windows
Server Explorer Toolbox
FinalProjectPart2B.cpp
FinalProjectPart2B (Global Scope)
#include <iostream>
#include <string>
using namespace std;
void selectionSort(string[], int);
void swap(string& a, string& b);
void displaySortedList(string[], int size);
int binarySearch(string[], int, string);
string getUserSearchString();
void displaySearchResults(string, int);
int main()
{
const int NUM_NAMES = 20; //array size
string names[NUM_NAMES] = { "Collins, Bill",
"Griffin, Jim",
"Taylor, Terri",
"Allison, Jeff",
"James, Jean",
"Rutherford, Gre
"Harrison, Rose"
"Pike, Gordon",
int results; //to hold th
string searchString; //to hold se
//sort the list
selectionSort(names, NUM_NAMES);
Microsoft Visual Studio Debug Console
Sorted Names (Ascending Order)
1. Allen, Jim
2. Allison, Jeff
3. Collins, Bill
4. Griffin, Jim
5. Harrison, Rose
6. Holland, Beth
7. James, Jean
8. Javens, Renee
9. Johnson, Jill
10. Looney, Joe
11. Pike, Gordon
12. Pore, Bob
13. Rose, Geri
14. Rutherford, Greg
15. Setzer, Cathy
16. Smith, Bart
17. Stamey, Marty
18. Taylor, Terri
19. Weaver, Jim
20. Wolfe, Bill
Please enter the name you would like to search for: Pore, Bob
Pore, Bob is number 12 in the list
and is located in the array at element 11.
C:\Users\blair\source\repos\FinalProjectPart2B\Debug\FinalProjectPart2B.e
32604) exited with code 0.
Press any key to close this window . . .