

```

// 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 Search

Debug x86 Local Windows Debugger

FinalProjectPart1Payroll.cpp

FinalProjectPart1Payroll (Global Scope)

```

1 // Mickie Blair
2 // CIST 2361 - CRN 63227
3 // Summer 2019 Final Project
4 // Part 1 Payroll
5 // Payroll program with four arrays to hold employee ID, hours worked,
6 // pay rate, and gross wages. The gross wages will be calculated and displayed
7 // pay rate and hours worked.
8
9 #include <iostream>
10 #include <iomanip>
11
12 using namespace std;
13
14 void displayIntro();
15 void getEmployeeData(long[], double[], double[], int);
16 void calcGrossWages(long[], double[], double[], double[], int);
17 void displayResults(long[], double[], int);
18
19 int main()
20 {
21
22     long empID[] = { 5658845, 4520125, 7895122,
23                     8777541, 8451277, 1302850,
24                     7580489}; //array for Employee ID
25
26     const int SIZE = 7; //constant to hold array size
27     double hours[SIZE]; //array for hours worked
28     double payRate[SIZE]; //array for hourly pay rate
29     double wages[SIZE]; //array for gross wages
30
31     //display introduction

```

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\Debug\exe (process 27028) exited with code 0.

100 % No issues found

Error List Output

Ready Ln 119

```

// 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;
}

```

The screenshot shows the Microsoft Visual Studio IDE with the file `FinalProjectPart2A.cpp` open. The code in the editor includes comments, headers, namespace declarations, and function definitions for a selection sort algorithm. The `main` function initializes an array of names and calls the `selectionSort` function. The Debug Console on the right displays the output of the program, showing the sorted names in ascending order.

```

1 // Mickie Blair
2 // CIST 2361 - CRN 63227
3 // Summer 2019 Final Project
4 // Part 2 - Selection Sort
5 // A. String Selection Sort
6
7 #include <iostream>
8 #include <string>
9
10 using namespace std;
11
12 void selectionSort(string[], int);
13 void swap(string& a, string& b);
14 void displaySortedList(string[], int size);
15
16 int main()
17 {
18     const int NUM_NAMES = 20;
19     string names[NUM_NAMES] = { "Collins, Bill",
20                                "Griffin, Jim",
21                                "Taylor, Terri",
22                                "Allison, Jeff",
23                                "James, Jean",
24                                "Rutherford, Gre",
25                                "Harrison, Rose",
26                                "Pike, Gordon",
27
28
29     //sort the list
30     selectionSort(names, NUM_NAMES);

```

Sorted Names (Ascending Order)

Allen, Jim  
Allison, Jeff  
Collins, Bill  
Griffin, Jim  
Harrison, Rose  
Holland, Beth  
James, Jean  
Javens, Renee  
Johnson, Jill  
Looney, Joe  
Pike, Gordon  
Pore, Bob  
Rose, Geri  
Rutherford, Greg  
Setzer, Cathy  
Smith, Bart  
Stamey, Marty  
Taylor, Terri  
Weaver, Jim  
Wolfe, Bill

C:\Users\blair\source\repos\Final ProjectPart2A\Debug (x86) exited with code 0.  
Press any key to close this window . . .

```

// 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

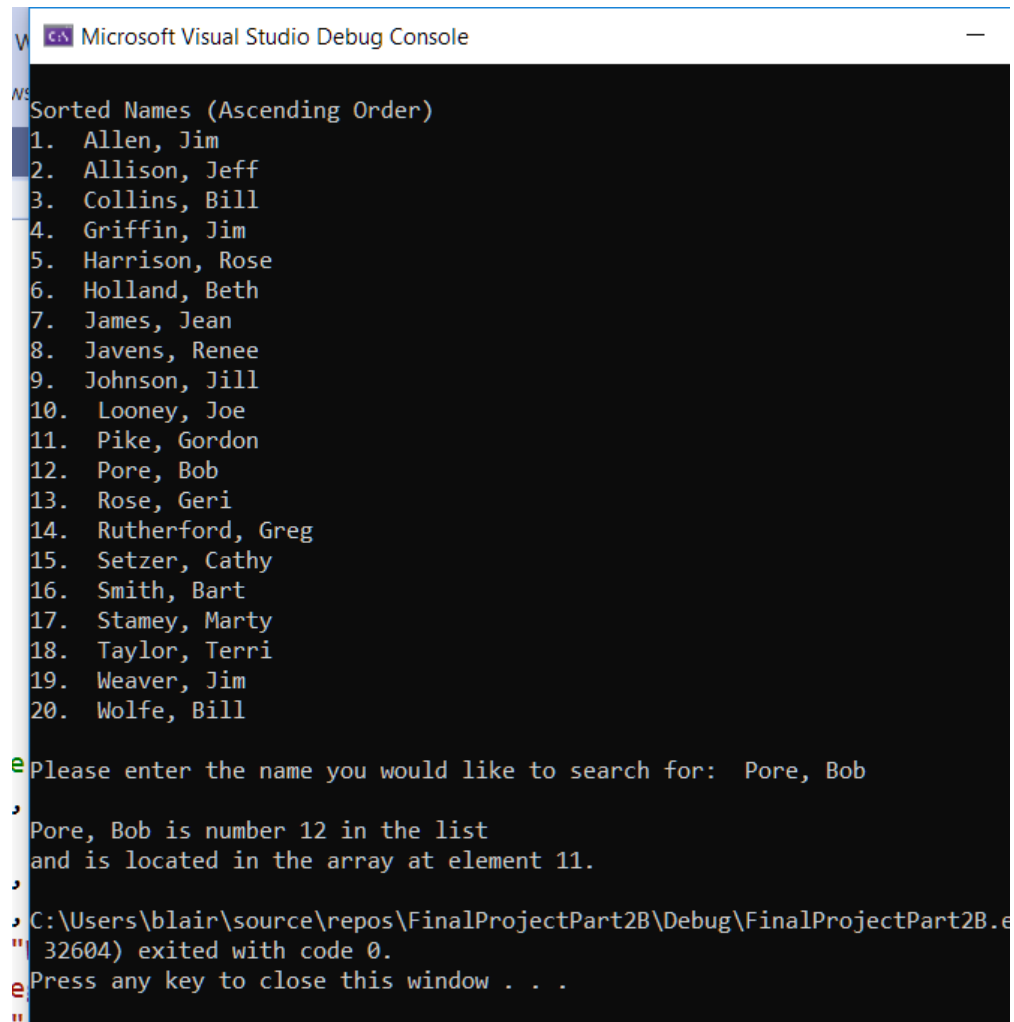
```

```

        {
            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";
    }
}

```



The screenshot shows the Microsoft Visual Studio Debug Console with a black background and white text. The output of the program is as follows:

```

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 . . .

```



The image shows a screenshot of the Microsoft Visual Studio IDE. The left pane displays the source code for `FinalProjectPart2B.cpp`. The code includes headers for `<iostream>` and `<string>`, uses the `std` namespace, and defines several functions: `selectionSort`, `swap`, `displaySortedList`, `binarySearch`, `getUserSearchString`, and `displaySearchResults`. The `main` function initializes an array of 20 names, sorts them using `selectionSort`, and then prompts the user for a search string. The right pane shows the Microsoft Visual Studio Debug Console. It displays the sorted names in ascending order, followed by the user input "Pore, Bob". The console then shows the search results: "Pore, Bob is number 12 in the list and is located in the array at element 11." The program then exits with code 0.

```
7 #include <iostream>
8 #include <string>
9
10 using namespace std;
11
12 void selectionSort(string[], int);
13 void swap(string& a, string& b);
14 void displaySortedList(string[], int size);
15 int binarySearch(string[], int, string);
16 string getUserSearchString();
17 void displaySearchResults(string, int);
18
19 int main()
20 {
21     const int NUM_NAMES = 20; //array size
22     string names[NUM_NAMES] = { "Collins, Bill",
23                                 "Griffin, Jim",
24                                 "Taylor, Terri",
25                                 "Allison, Jeff",
26                                 "James, Jean",
27                                 "Rutherford, Greg",
28                                 "Harrison, Rose",
29                                 "Pike, Gordon",
30
31     int results; //to hold the results
32     string searchString; //to hold the search string
33
34
35     //sort the list
36     selectionSort(names, NUM_NAMES);
37
38     //get the search string
39     searchString = getUserSearchString();
40
41     //display the sorted list
42     displaySortedList(names, NUM_NAMES);
43
44     //display the search results
45     displaySearchResults(searchString, NUM_NAMES);
46 }
```

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.exe (32604) exited with code 0.  
Press any key to close this window . . .