Muhammad Mufeez

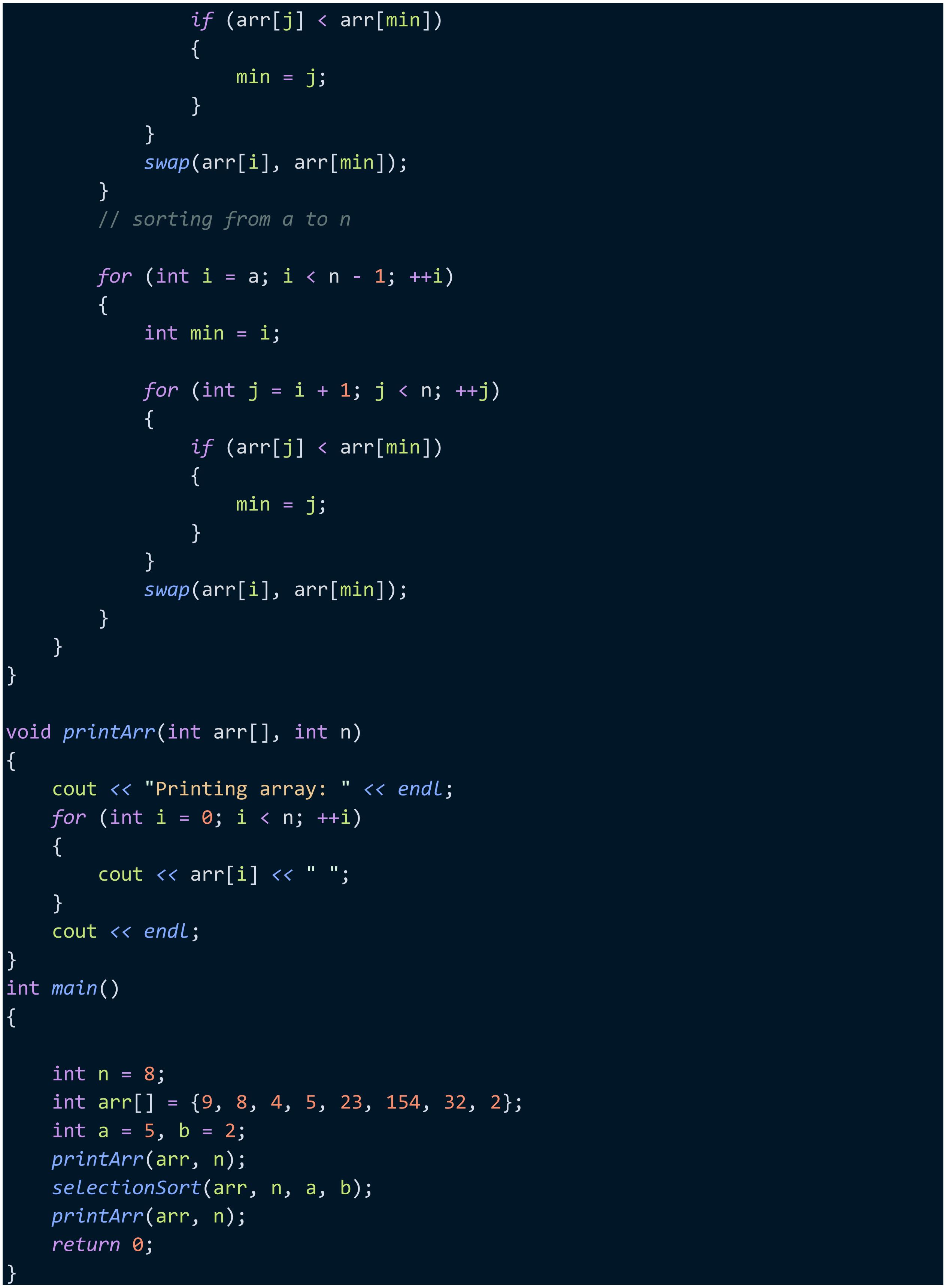
BCS-3J

DS Lab 04

Q1:

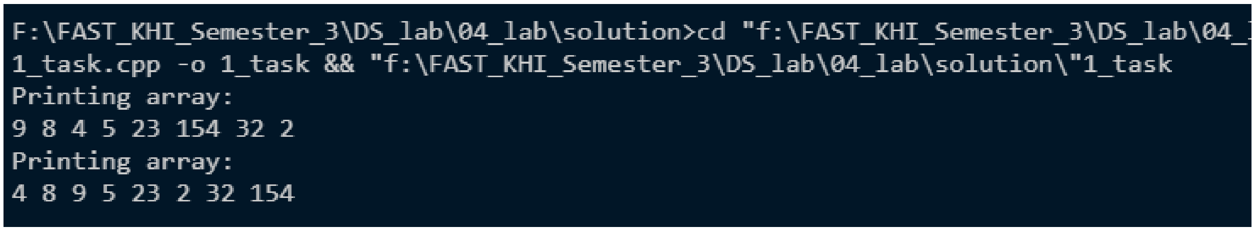
Code:

```
#include <bits/stdc++.h>

using namespace std;

void selectionSort(int arr[], int n, int a, int b)
{
    if (a == b)
        return;
    else if (b > a)
    {
        for (int i = a; i < b; ++i)
        {
            int min = i;

            for (int j = i + 1; j < b + 1; ++j)
            {
                if (arr[j] < arr[min])
                {
                    min = j;
                }
            }
            swap(arr[i], arr[min]);
        }
    }
    else
    {
        // sorting from 0 to b
        for (int i = 0; i < b; ++i)
        {
            int min = i;

            for (int j = i + 1; j < b + 1; ++j)
            {
```



```
                if (arr[j] < arr[min])
                {
                    min = j;
                }
            }
            swap(arr[i], arr[min]);
        }
        // sorting from a to n

        for (int i = a; i < n - 1; ++i)
        {
            int min = i;

            for (int j = i + 1; j < n; ++j)
            {
                if (arr[j] < arr[min])
                {
                    min = j;
                }
            }
            swap(arr[i], arr[min]);
        }
    }
}

void printArr(int arr[], int n)
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}
int main()
{

    int n = 8;
    int arr[] = {9, 8, 4, 5, 23, 154, 32, 2};
    int a = 5, b = 2;
    printArr(arr, n);
    selectionSort(arr, n, a, b);
    printArr(arr, n);
    return 0;
}
```

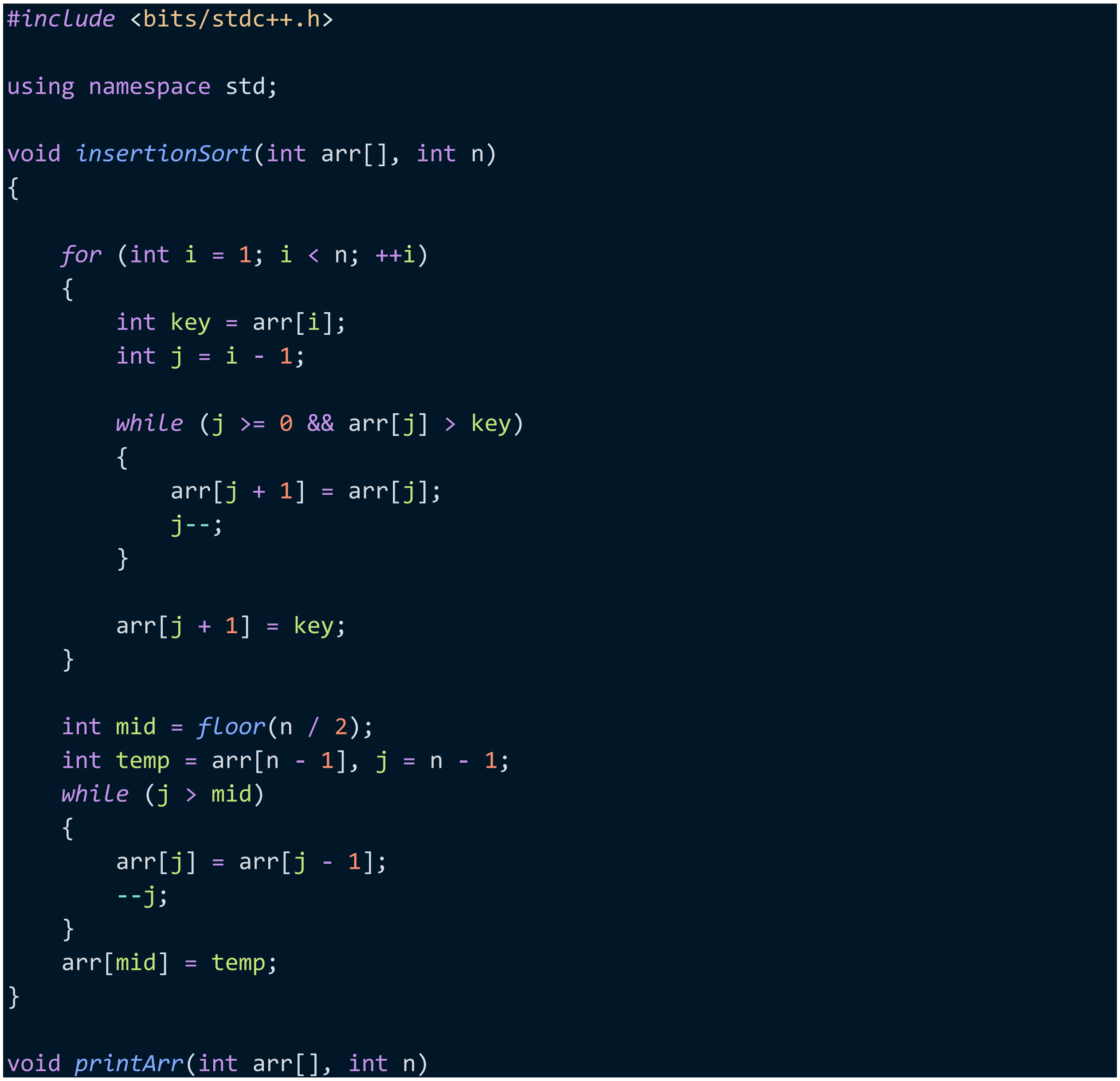
Output:



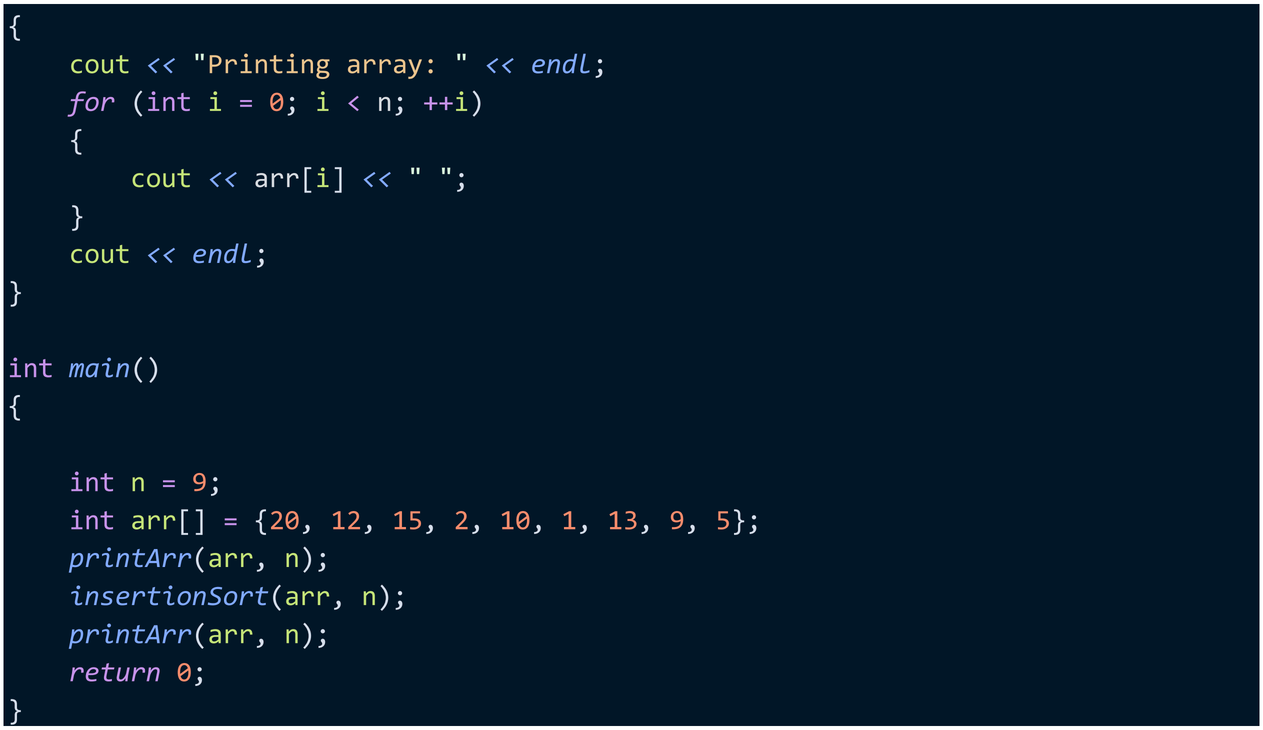
```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>cd "f:\FAST_KHI_Semester_3\DS_lab\04_
1_task.cpp -o 1_task && "f:\FAST_KHI_Semester_3\DS_lab\04_lab\solution\"1_task
Printing array:
9 8 4 5 23 154 32 2
Printing array:
4 8 9 5 23 2 32 154
```

Q2:

Code:

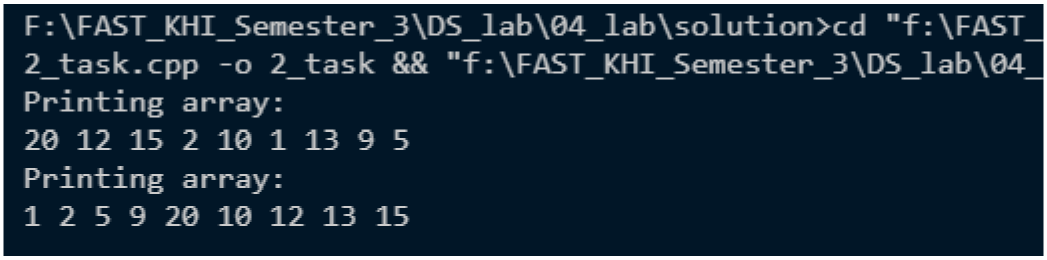


```cpp
#include <bits/stdc++.h>

using namespace std;

void insertionSort(int arr[], int n)
{

    for (int i = 1; i < n; ++i)
    {
        int key = arr[i];
        int j = i - 1;

        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j--;
        }

        arr[j + 1] = key;
    }

    int mid = floor(n / 2);
    int temp = arr[n - 1], j = n - 1;
    while (j > mid)
    {
        arr[j] = arr[j - 1];
        --j;
    }
    arr[mid] = temp;
}

void printArr(int arr[], int n)
```



```
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{

    int n = 9;
    int arr[] = {20, 12, 15, 2, 10, 1, 13, 9, 5};
    printArr(arr, n);
    insertionSort(arr, n);
    printArr(arr, n);
    return 0;
}
```

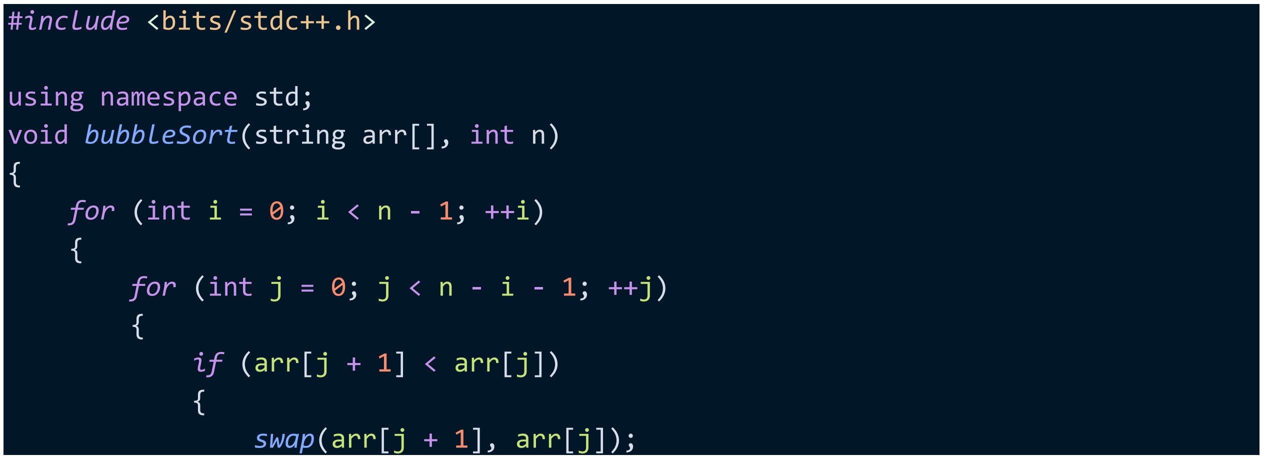
Output:



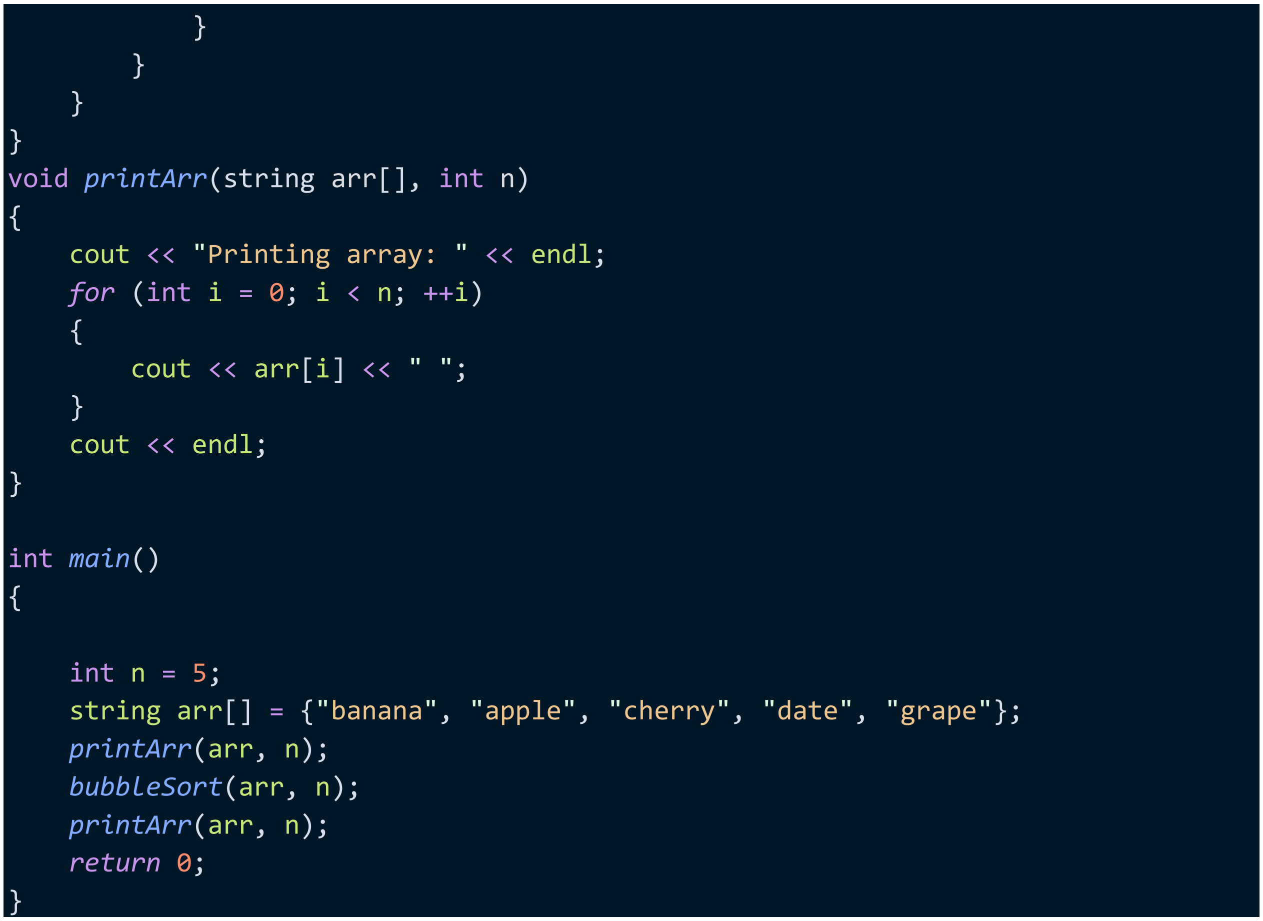
```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>cd "f:\FAST_
2_task.cpp -o 2_task && "f:\FAST_KHI_Semester_3\DS_lab\04_
Printing array:
20 12 15 2 10 1 13 9 5
Printing array:
1 2 5 9 20 10 12 13 15
```

Q3:

Code:

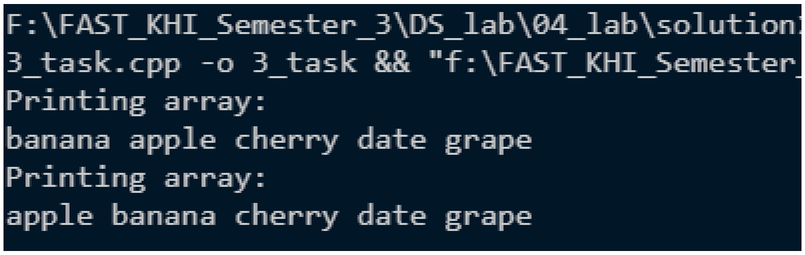


```
#include <bits/stdc++.h>

using namespace std;
void bubbleSort(string arr[], int n)
{
    for (int i = 0; i < n - 1; ++i)
    {
        for (int j = 0; j < n - i - 1; ++j)
        {
            if (arr[j + 1] < arr[j])
            {
                swap(arr[j + 1], arr[j]);
```



```
            }
        }
    }
}
void printArr(string arr[], int n)
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{

    int n = 5;
    string arr[] = {"banana", "apple", "cherry", "date", "grape"};
    printArr(arr, n);
    bubbleSort(arr, n);
    printArr(arr, n);
    return 0;
}
```

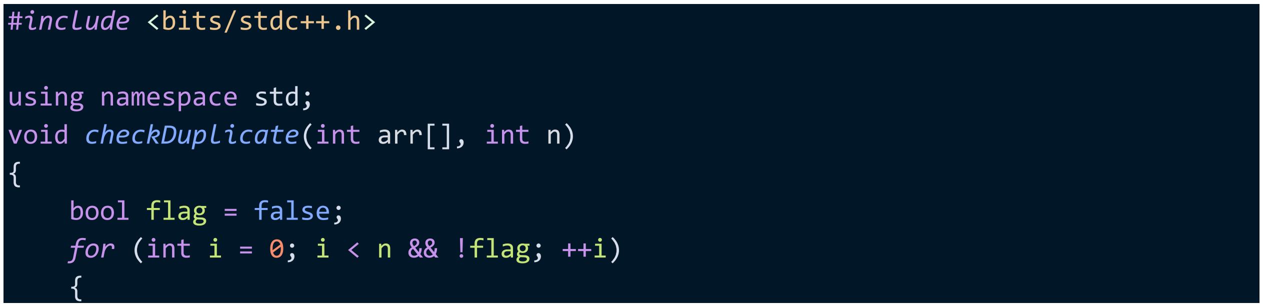
Output:



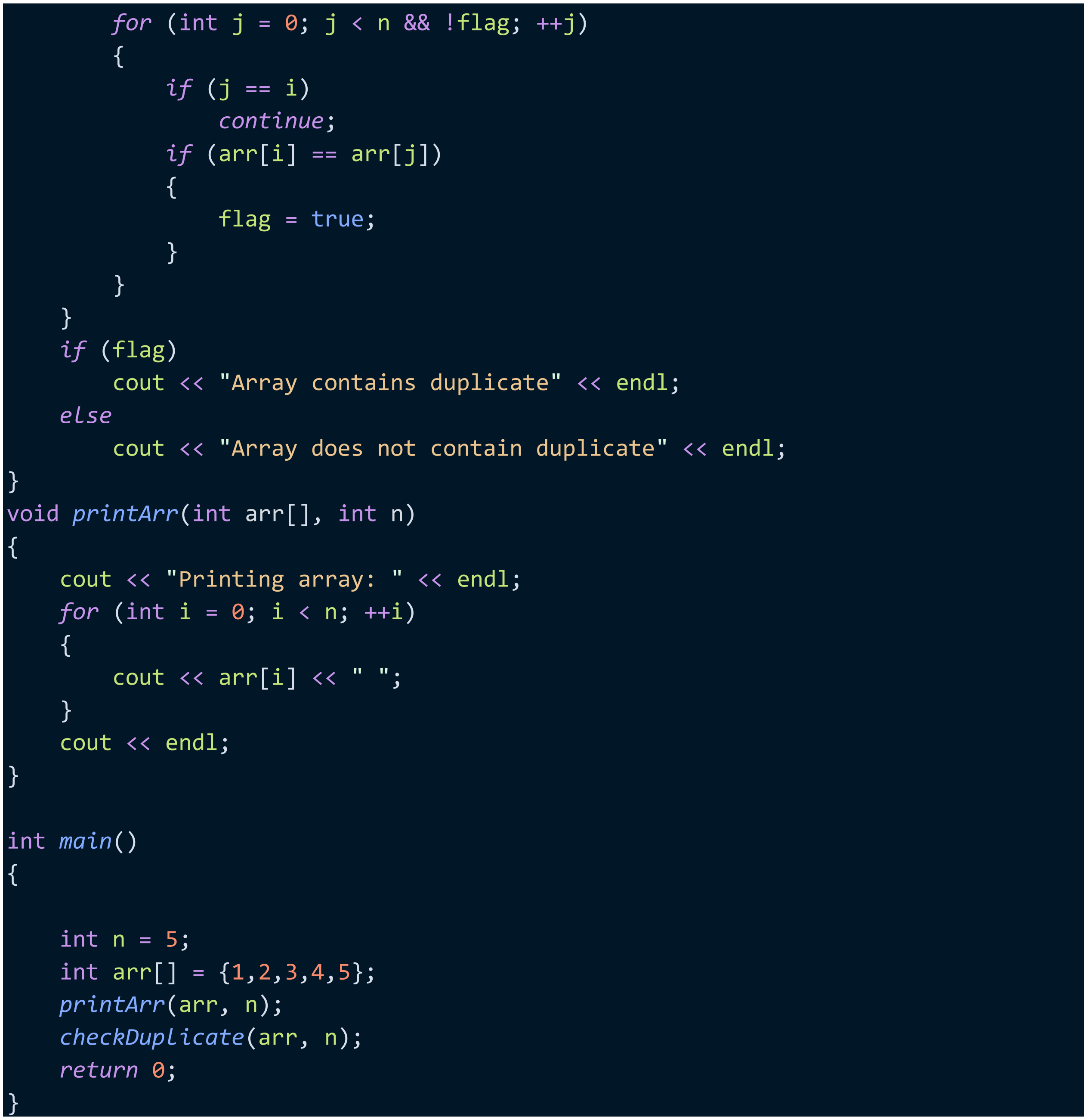
```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution
3_task.cpp -o 3_task && "f:\FAST_KHI_Semester_
Printing array:
banana apple cherry date grape
Printing array:
apple banana cherry date grape
```

Q4:

Code:

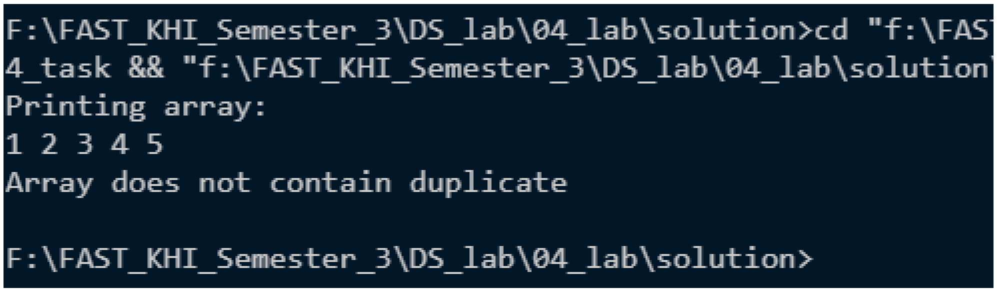


```
#include <bits/stdc++.h>

using namespace std;
void checkDuplicate(int arr[], int n)
{
    bool flag = false;
    for (int i = 0; i < n && !flag; ++i)
    {
```



```
        for (int j = 0; j < n && !flag; ++j)
        {
            if (j == i)
                continue;
            if (arr[i] == arr[j])
            {
                flag = true;
            }
        }
    }
    if (flag)
        cout << "Array contains duplicate" << endl;
    else
        cout << "Array does not contain duplicate" << endl;
}
void printArr(int arr[], int n)
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{

    int n = 5;
    int arr[] = {1,2,3,4,5};
    printArr(arr, n);
    checkDuplicate(arr, n);
    return 0;
}
```

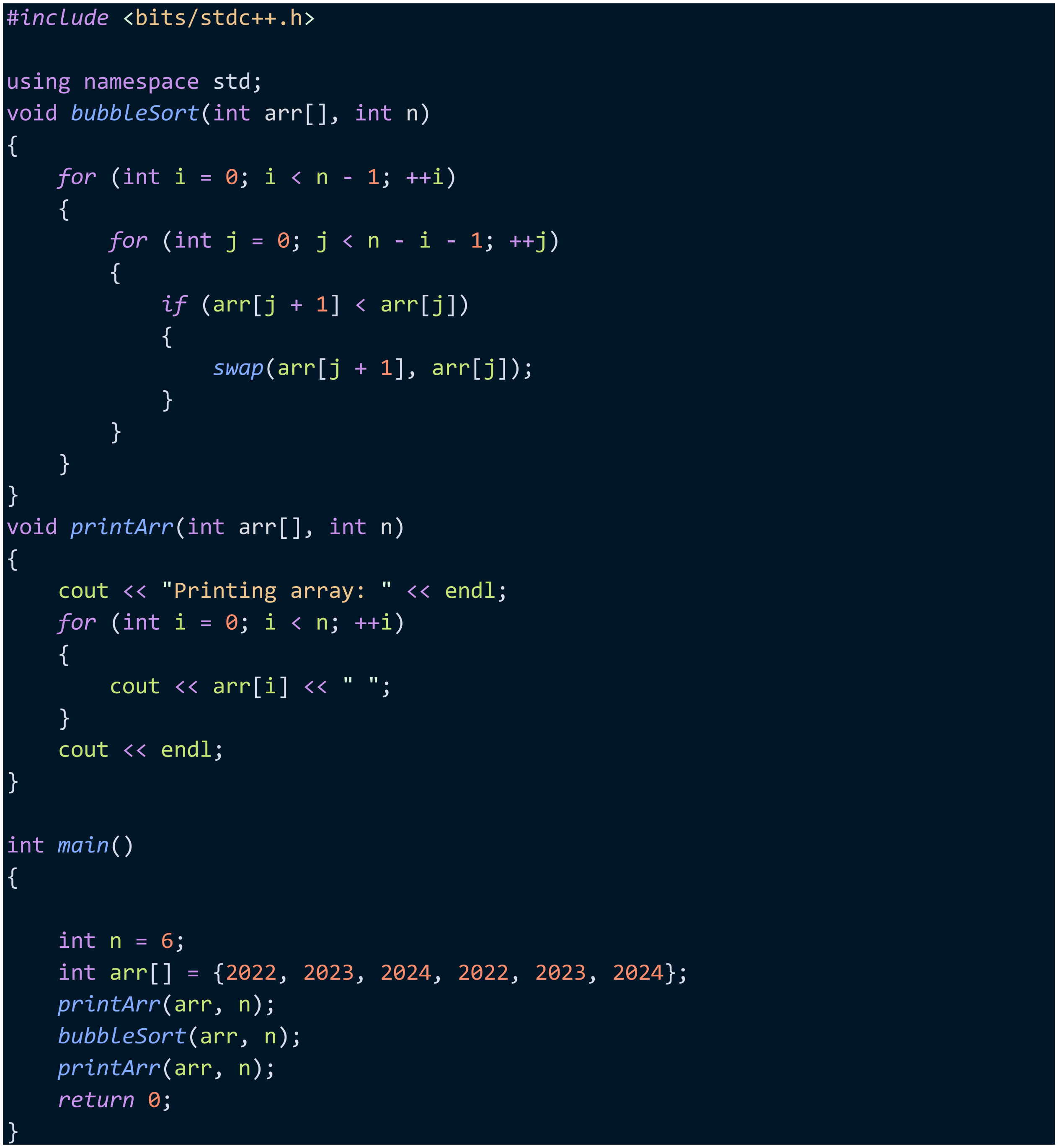
Output:



```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>cd "f:\FAS
4_task && "f:\FAST_KHI_Semester_3\DS_lab\04_lab\solution
Printing array:
1 2 3 4 5
Array does not contain duplicate

F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>
```

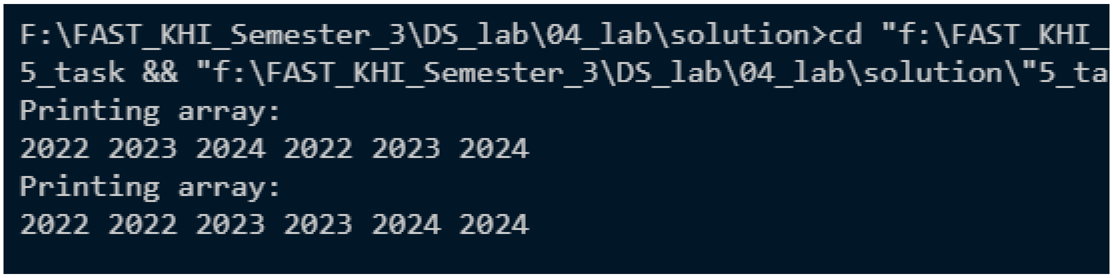
Q5:

Code:



```cpp
#include <bits/stdc++.h>

using namespace std;
void bubbleSort(int arr[], int n)
{
    for (int i = 0; i < n - 1; ++i)
    {
        for (int j = 0; j < n - i - 1; ++j)
        {
            if (arr[j + 1] < arr[j])
            {
                swap(arr[j + 1], arr[j]);
            }
        }
    }
}
void printArr(int arr[], int n)
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{

    int n = 6;
    int arr[] = {2022, 2023, 2024, 2022, 2023, 2024};
    printArr(arr, n);
    bubbleSort(arr, n);
    printArr(arr, n);
    return 0;
}
```

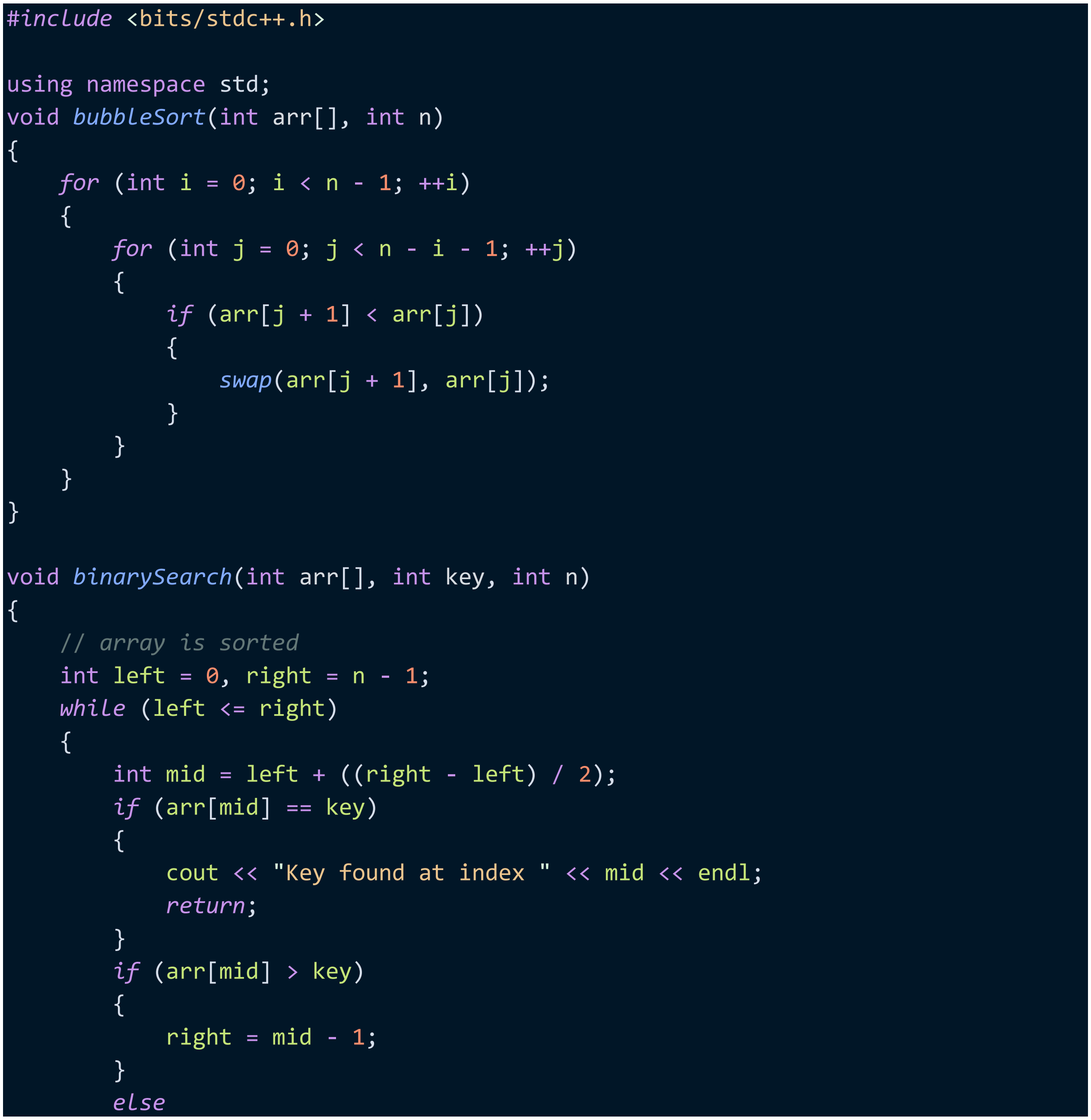
Output:



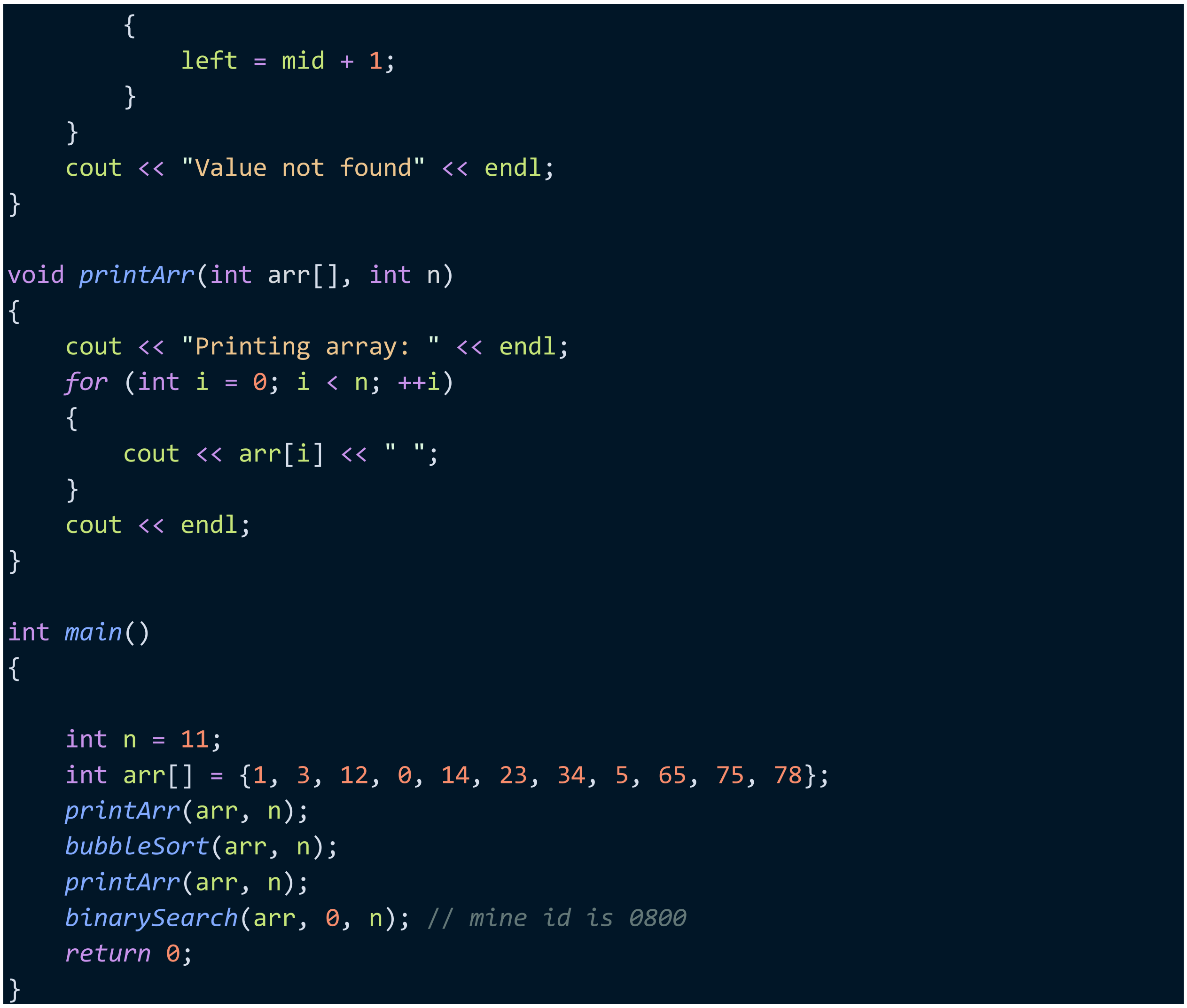
```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>cd "f:\FAST_KHI_
5_task && "f:\FAST_KHI_Semester_3\DS_lab\04_lab\solution\"5_ta
Printing array:
2022 2023 2024 2022 2023 2024
Printing array:
2022 2022 2023 2023 2024 2024
```

Q6:

Code:

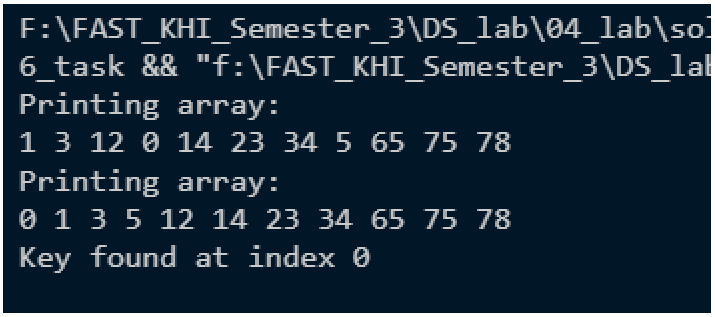


```
#include <bits/stdc++.h>

using namespace std;
void bubbleSort(int arr[], int n)
{
    for (int i = 0; i < n - 1; ++i)
    {
        for (int j = 0; j < n - i - 1; ++j)
        {
            if (arr[j + 1] < arr[j])
            {
                swap(arr[j + 1], arr[j]);
            }
        }
    }
}

void binarySearch(int arr[], int key, int n)
{
    // array is sorted
    int left = 0, right = n - 1;
    while (left <= right)
    {
        int mid = left + ((right - left) / 2);
        if (arr[mid] == key)
        {
            cout << "Key found at index " << mid << endl;
            return;
        }
        if (arr[mid] > key)
        {
            right = mid - 1;
        }
        else
```



```
        {
            left = mid + 1;
        }
    }
    cout << "Value not found" << endl;
}

void printArr(int arr[], int n)
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{

    int n = 11;
    int arr[] = {1, 3, 12, 0, 14, 23, 34, 5, 65, 75, 78};
    printArr(arr, n);
    bubbleSort(arr, n);
    printArr(arr, n);
    binarySearch(arr, 0, n); // mine id is 0800
    return 0;
}
```

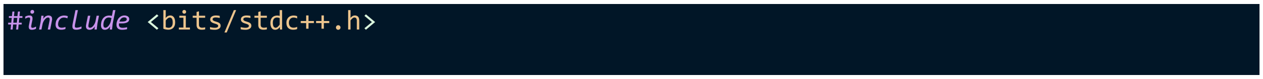
Output:



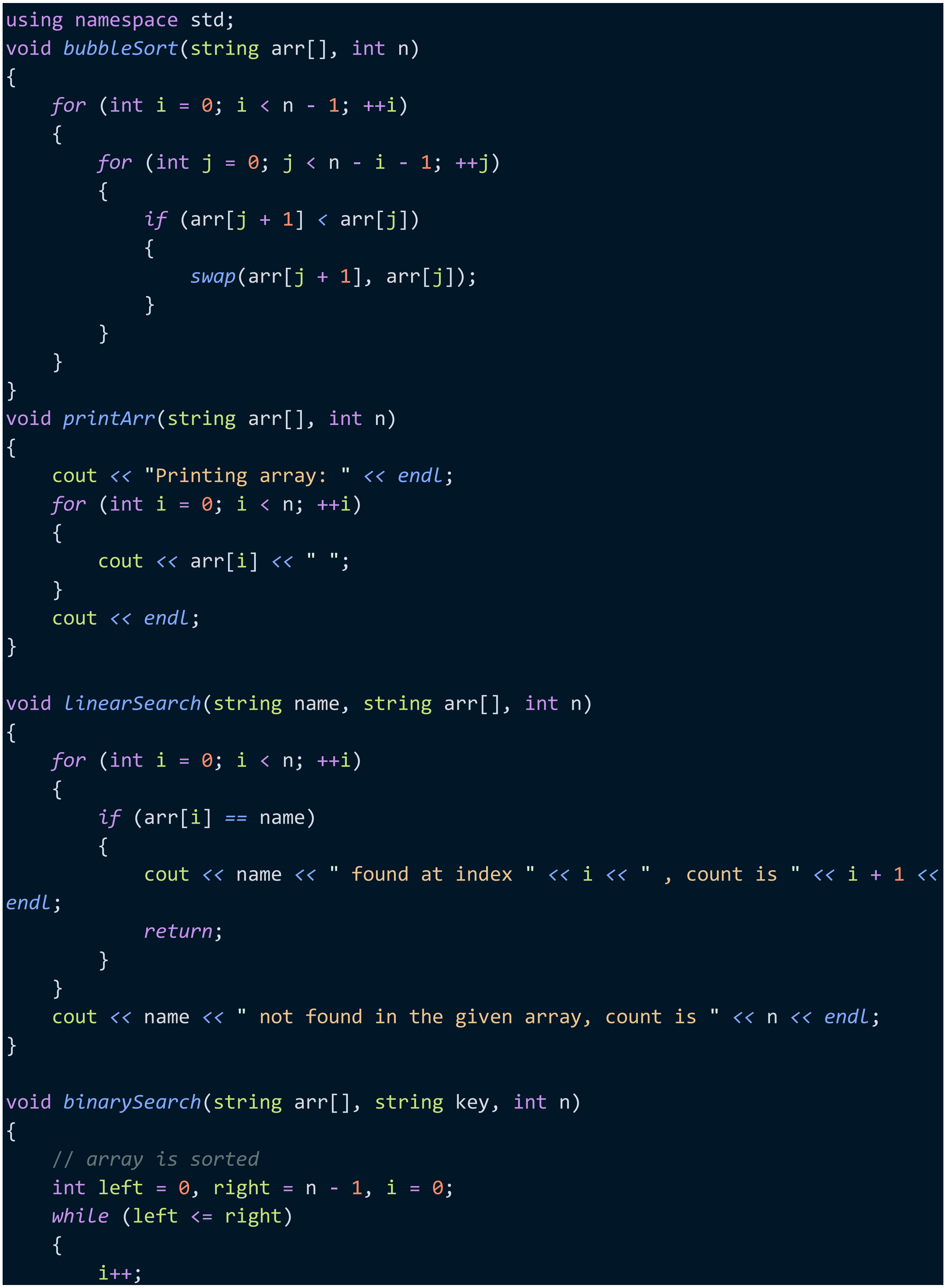
```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\so
6_task && "f:\FAST_KHI_Semester_3\DS_la
Printing array:
1 3 12 0 14 23 34 5 65 75 78
Printing array:
0 1 3 5 12 14 23 34 65 75 78
Key found at index 0
```

Q7:

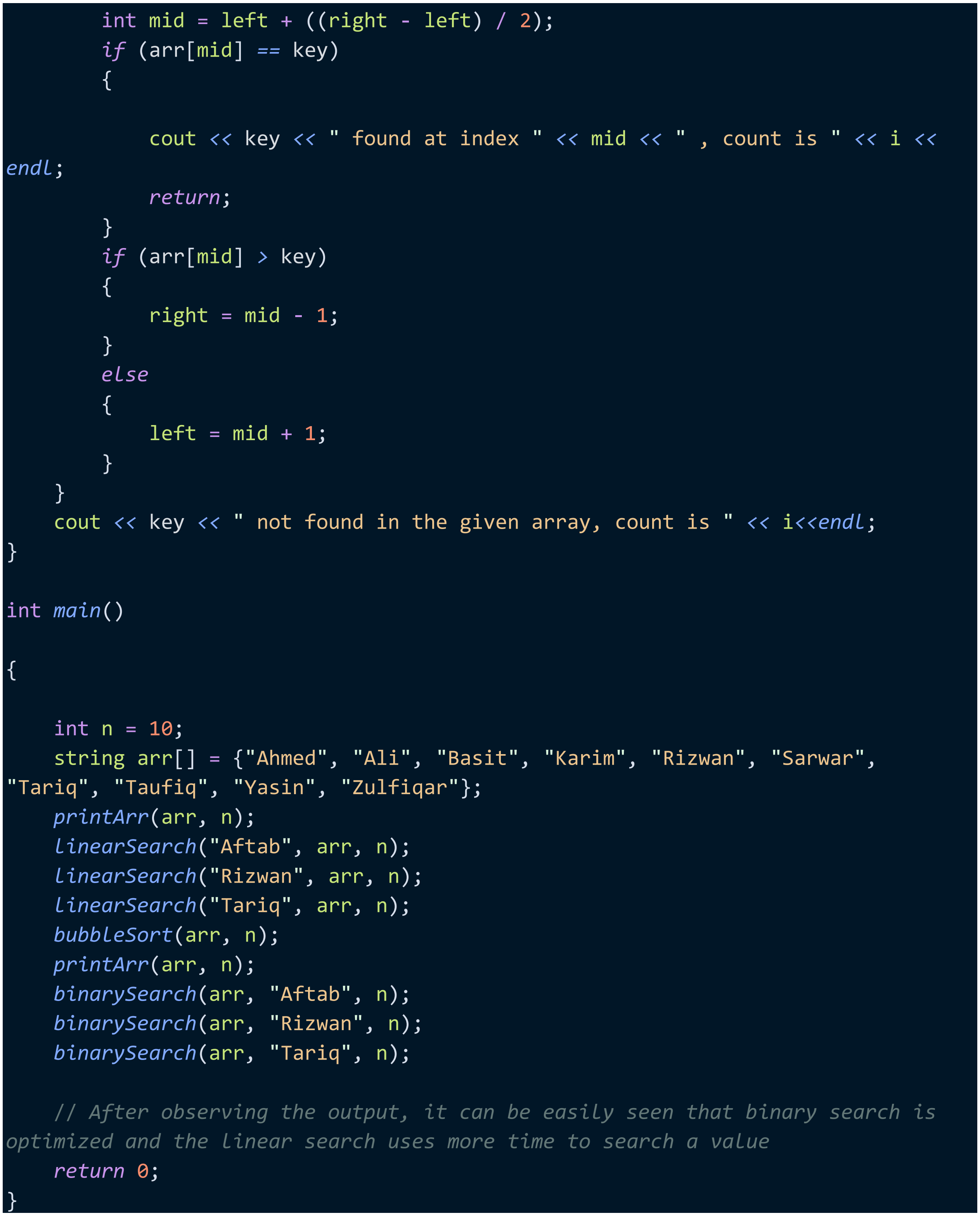
Code:



```
#include <bits/stdc++.h>
```

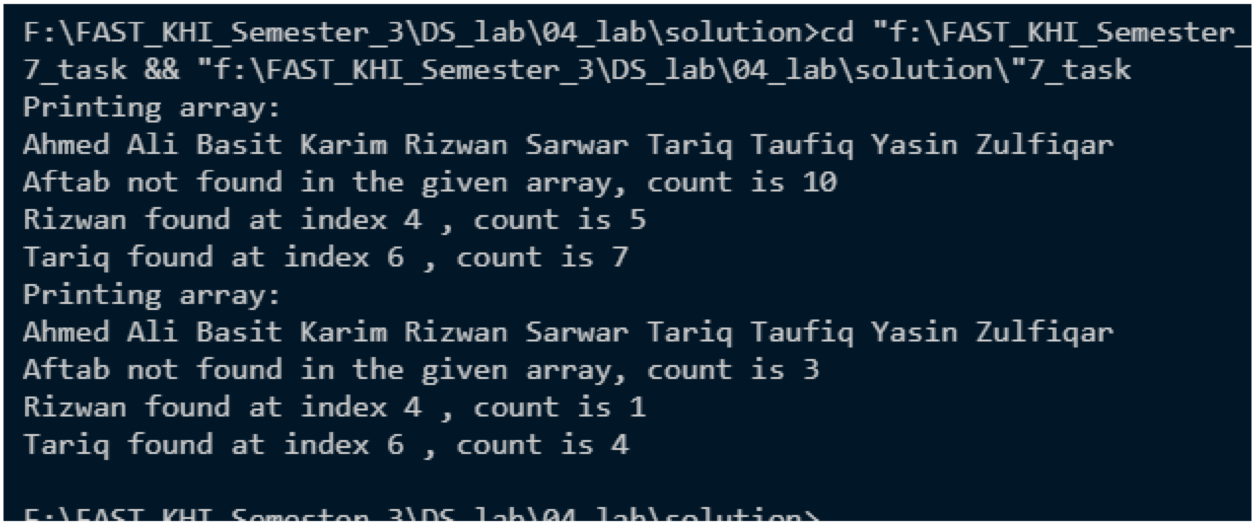


```
using namespace std;
void bubbleSort(string arr[], int n)
{
    for (int i = 0; i < n - 1; ++i)
    {
        for (int j = 0; j < n - i - 1; ++j)
        {
            if (arr[j + 1] < arr[j])
            {
                swap(arr[j + 1], arr[j]);
            }
        }
    }
}
void printArr(string arr[], int n)
{
    cout << "Printing array: " << endl;
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

void linearSearch(string name, string arr[], int n)
{
    for (int i = 0; i < n; ++i)
    {
        if (arr[i] == name)
        {
            cout << name << " found at index " << i << " , count is " << i + 1 <<
endl;
            return;
        }
    }
    cout << name << " not found in the given array, count is " << n << endl;
}

void binarySearch(string arr[], string key, int n)
{
    // array is sorted
    int left = 0, right = n - 1, i = 0;
    while (left <= right)
    {
        i++;
```



```
        int mid = left + ((right - left) / 2);
        if (arr[mid] == key)
        {

            cout << key << " found at index " << mid << " , count is " << i <<
endl;
            return;
        }
        if (arr[mid] > key)
        {
            right = mid - 1;
        }
        else
        {
            left = mid + 1;
        }
    }
    cout << key << " not found in the given array, count is " << i<<endl;
}

int main()

{

    int n = 10;
    string arr[] = {"Ahmed", "Ali", "Basit", "Karim", "Rizwan", "Sarwar",
"Tariq", "Taufiq", "Yasin", "Zulfiqar"};
    printArr(arr, n);
    linearSearch("Aftab", arr, n);
    linearSearch("Rizwan", arr, n);
    linearSearch("Tariq", arr, n);
    bubbleSort(arr, n);
    printArr(arr, n);
    binarySearch(arr, "Aftab", n);
    binarySearch(arr, "Rizwan", n);
    binarySearch(arr, "Tariq", n);

    // After observing the output, it can be easily seen that binary search is
optimized and the linear search uses more time to search a value
    return 0;
}
```

Output:



```
F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>cd "f:\FAST_KHI_Semester_
7_task && "f:\FAST_KHI_Semester_3\DS_lab\04_lab\solution\"7_task
Printing array:
Ahmed Ali Basit Karim Rizwan Sarwar Tariq Taufiq Yasin Zulfiqar
Aftab not found in the given array, count is 10
Rizwan found at index 4 , count is 5
Tariq found at index 6 , count is 7
Printing array:
Ahmed Ali Basit Karim Rizwan Sarwar Tariq Taufiq Yasin Zulfiqar
Aftab not found in the given array, count is 3
Rizwan found at index 4 , count is 1
Tariq found at index 6 , count is 4

F:\FAST_KHI_Semester_3\DS_lab\04_lab\solution>
```