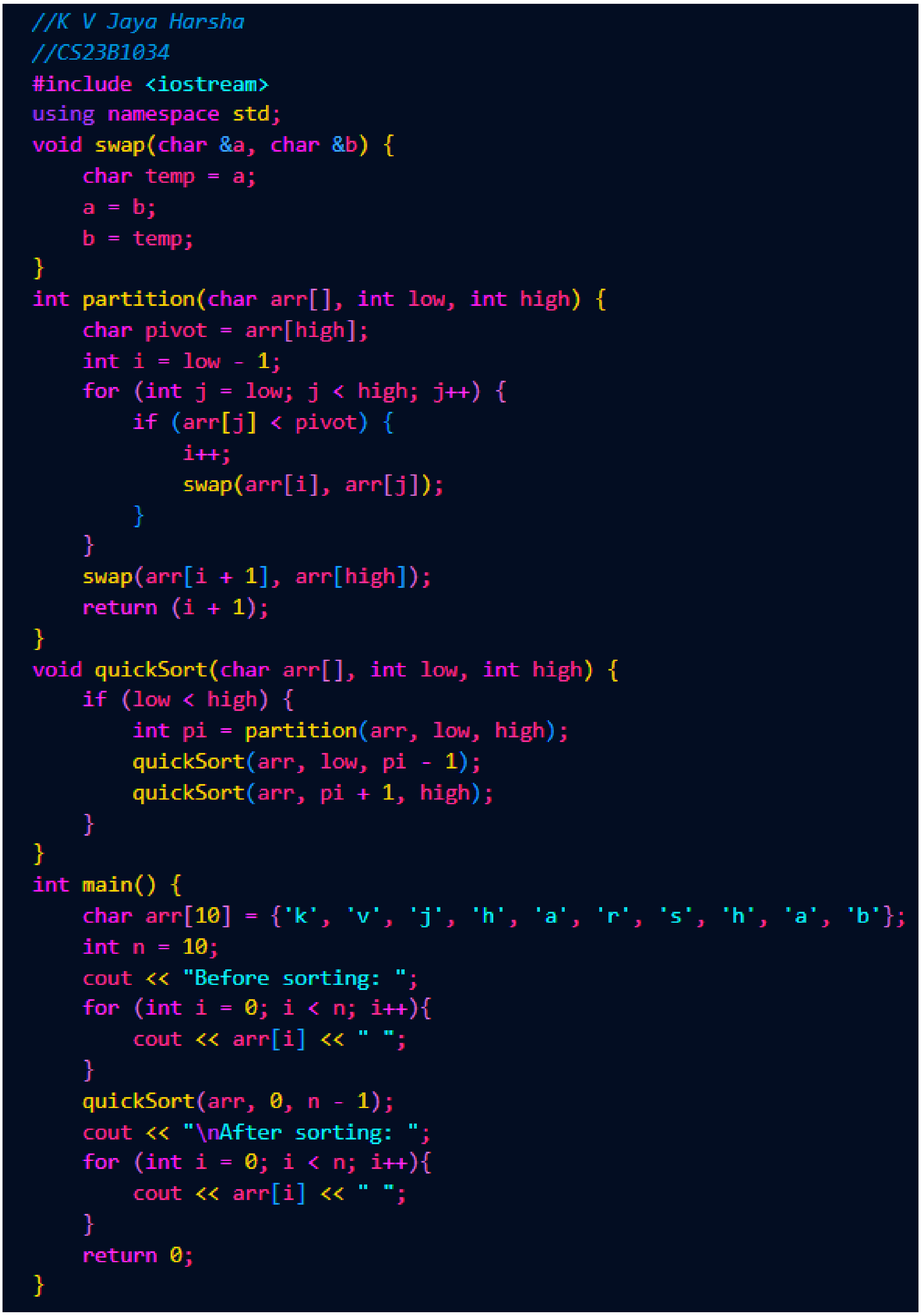
LAB-15

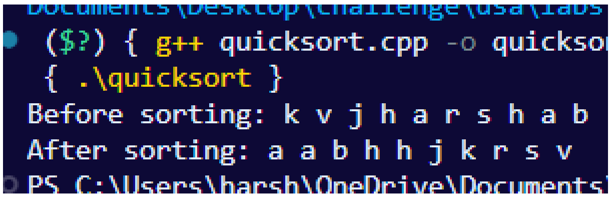
Name: K V Jaya Harsha

Roll no: CS23B1034 Date: 13-11-2024

Q1. Quicksort (in cpp)

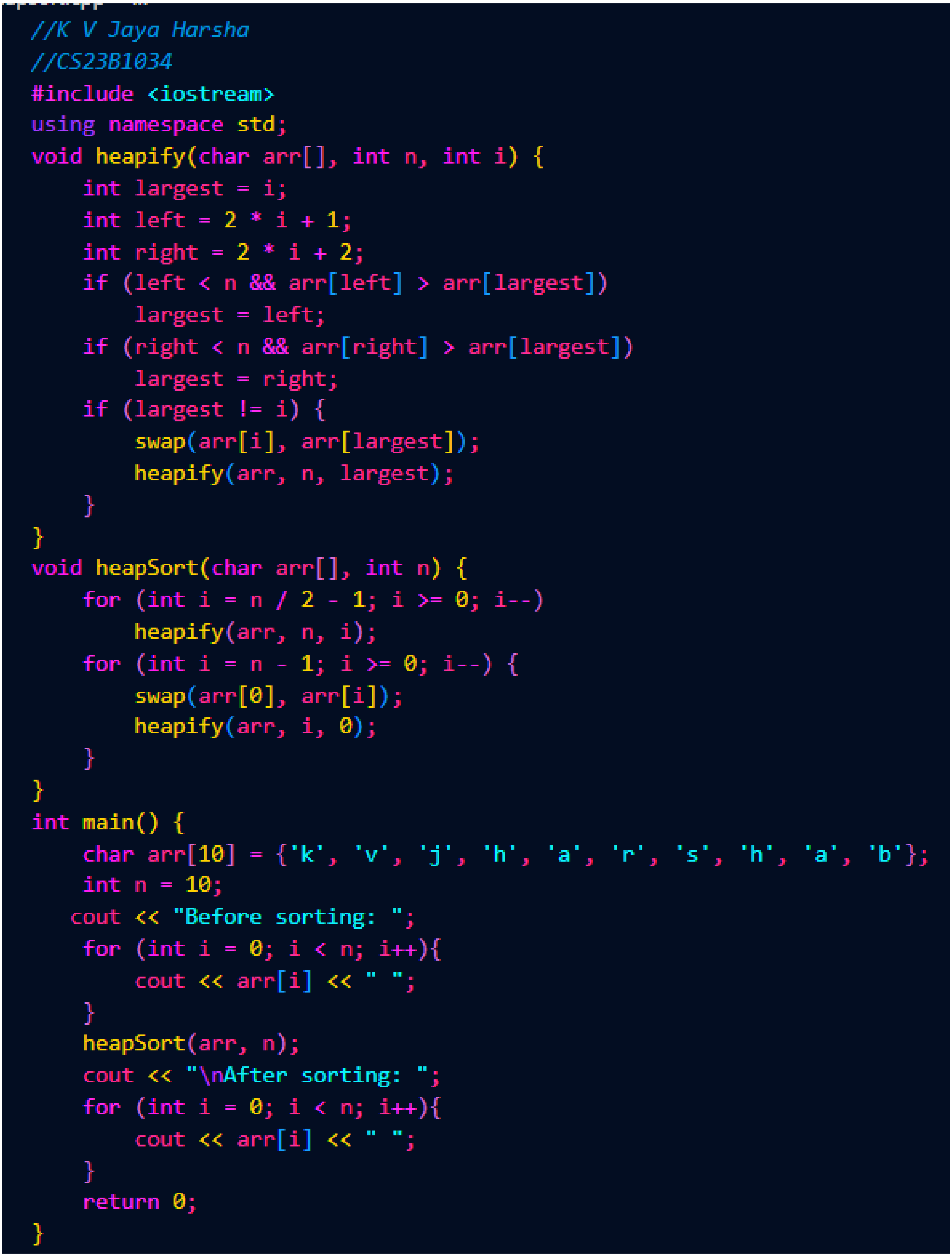


```cpp
//K V Jaya Harsha
//CS23B1034
#include <iostream>
using namespace std;
void swap(char &a, char &b) {
    char temp = a;
    a = b;
    b = temp;
}
int partition(char arr[], int low, int high) {
    char pivot = arr[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[i + 1], arr[high]);
    return (i + 1);
}
void quickSort(char arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
int main() {
    char arr[10] = {'k', 'v', 'j', 'h', 'a', 'r', 's', 'h', 'a', 'b'};
    int n = 10;
    cout << "Before sorting: ";
    for (int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    quickSort(arr, 0, n - 1);
    cout << "\nAfter sorting: ";
    for (int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    return 0;
}
```

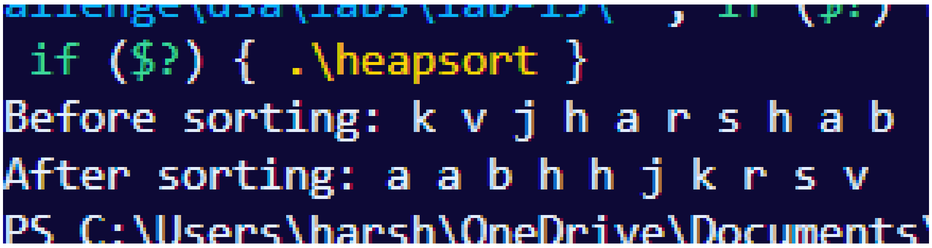


```
Documents\Desktop\challenge\dsa\labs
($?) { g++ quicksort.cpp -o quickso
{ .\quicksort }
Before sorting: k v j h a r s h a b
After sorting: a a b h h j k r s v
PS C:\Users\harsh\OneDrive\Documents\
```

Q2. Heapsort (in cpp)

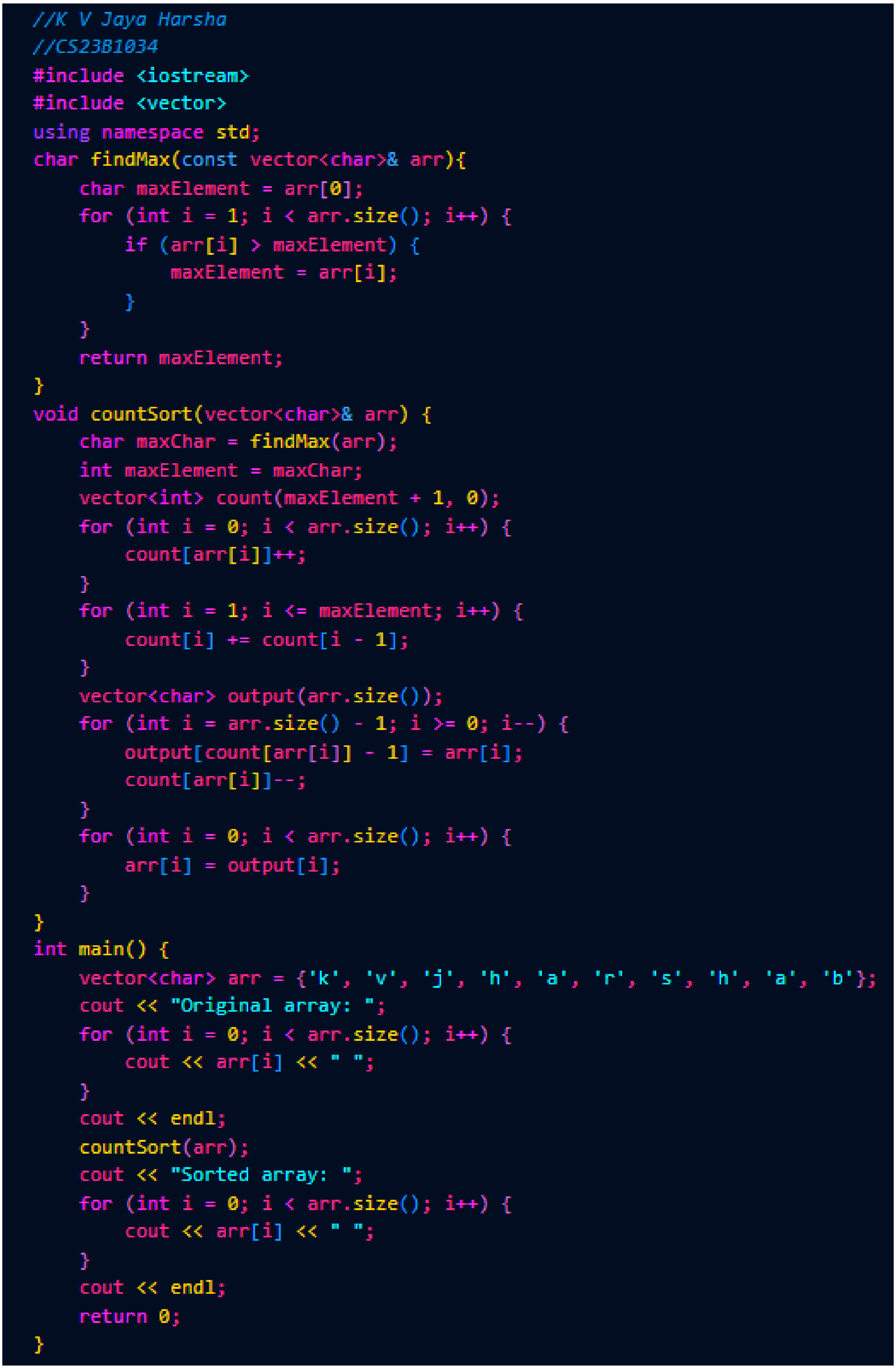


```cpp
//K V Jaya Harsha
//CS23B1034
#include <iostream>
using namespace std;
void heapify(char arr[], int n, int i) {
    int largest = i;
    int left = 2 * i + 1;
    int right = 2 * i + 2;
    if (left < n && arr[left] > arr[largest])
        largest = left;
    if (right < n && arr[right] > arr[largest])
        largest = right;
    if (largest != i) {
        swap(arr[i], arr[largest]);
        heapify(arr, n, largest);
    }
}
void heapSort(char arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--)
        heapify(arr, n, i);
    for (int i = n - 1; i >= 0; i--) {
        swap(arr[0], arr[i]);
        heapify(arr, i, 0);
    }
}
int main() {
    char arr[10] = {'k', 'v', 'j', 'h', 'a', 'r', 's', 'h', 'a', 'b'};
    int n = 10;
   cout << "Before sorting: ";
    for (int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    heapSort(arr, n);
    cout << "\nAfter sorting: ";
    for (int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    return 0;
}
```

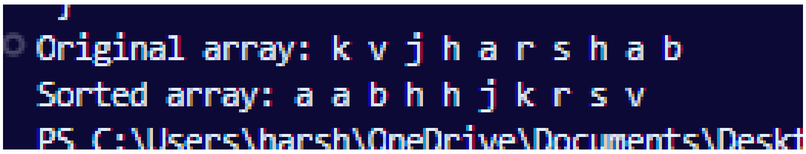


```
allenge\dsa\labs\lab-13\" ; if ($?)
 if ($?) { .\heapsort }
Before sorting: k v j h a r s h a b
After sorting: a a b h h j k r s v
PS C:\Users\harsh\OneDrive\Documents\
```

Q3. Count sort (in cpp)



```cpp
//K V Jaya Harsha
//CS23B1034
#include <iostream>
#include <vector>
using namespace std;
char findMax(const vector<char>& arr){
    char maxElement = arr[0];
    for (int i = 1; i < arr.size(); i++) {
        if (arr[i] > maxElement) {
            maxElement = arr[i];
        }
    }
    return maxElement;
}
void countSort(vector<char>& arr) {
    char maxChar = findMax(arr);
    int maxElement = maxChar;
    vector<int> count(maxElement + 1, 0);
    for (int i = 0; i < arr.size(); i++) {
        count[arr[i]]++;
    }
    for (int i = 1; i <= maxElement; i++) {
        count[i] += count[i - 1];
    }
    vector<char> output(arr.size());
    for (int i = arr.size() - 1; i >= 0; i--) {
        output[count[arr[i]] - 1] = arr[i];
        count[arr[i]]--;
    }
    for (int i = 0; i < arr.size(); i++) {
        arr[i] = output[i];
    }
}
int main() {
    vector<char> arr = {'k', 'v', 'j', 'h', 'a', 'r', 's', 'h', 'a', 'b'};
    cout << "Original array: ";
    for (int i = 0; i < arr.size(); i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    countSort(arr);
    cout << "Sorted array: ";
    for (int i = 0; i < arr.size(); i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```



```
Original array: k v j h a r s h a b
Sorted array: a a b h h j k r s v
PS C:\Users\harsh\OneDrive\Documents\Deskt
```