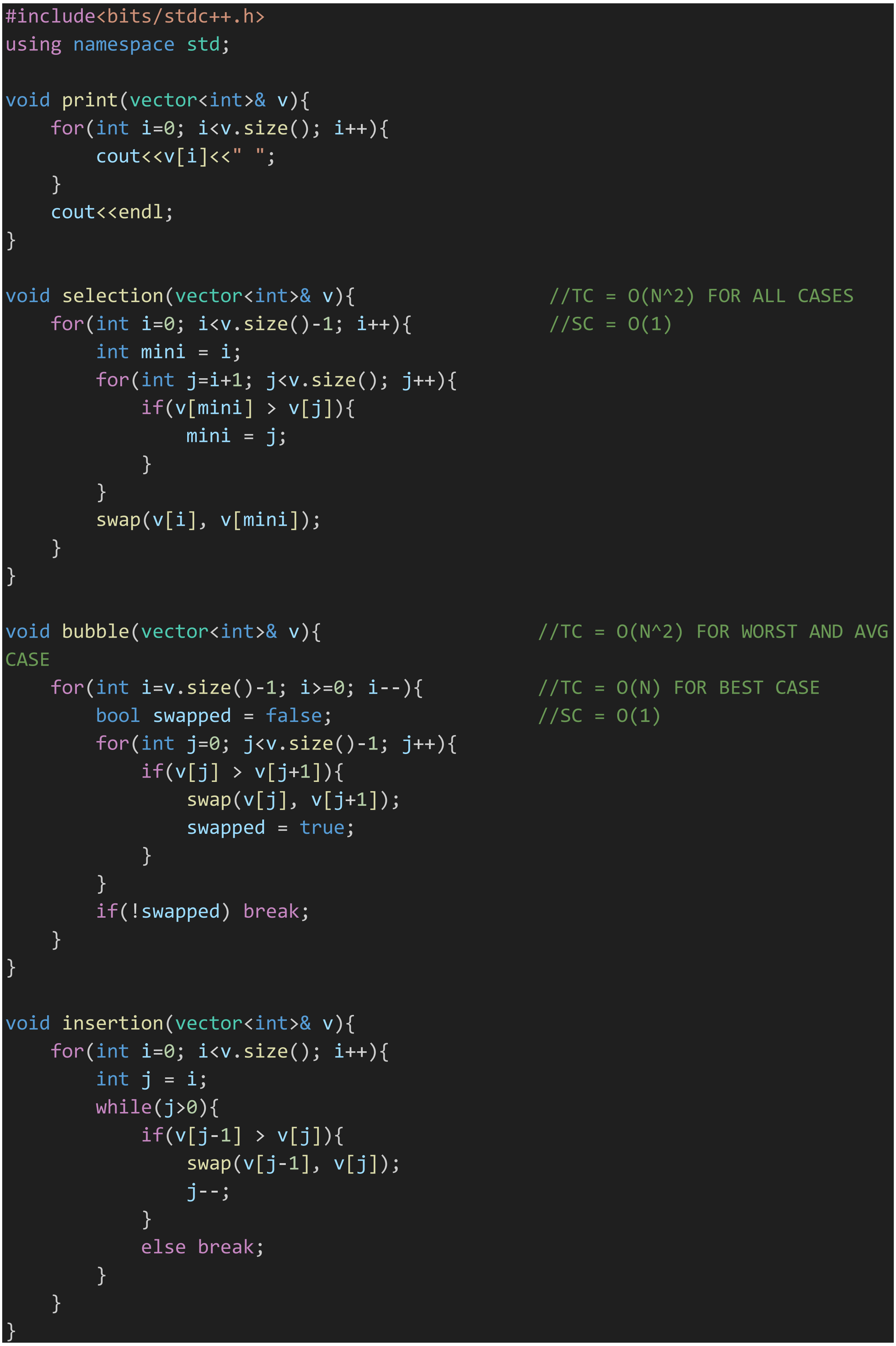
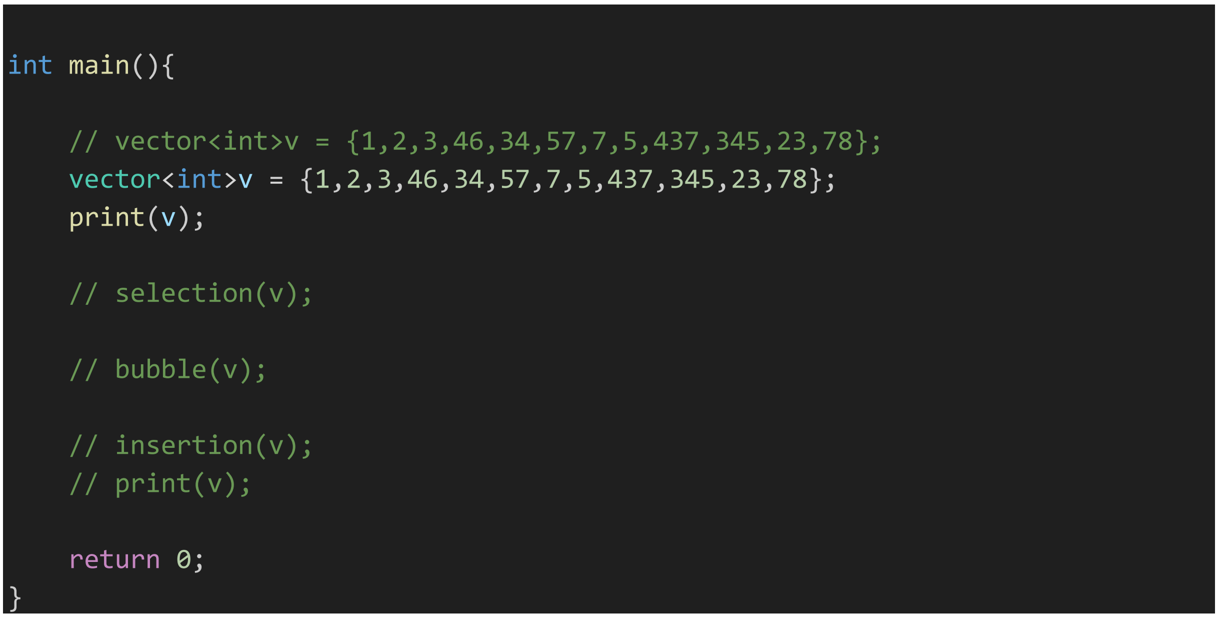
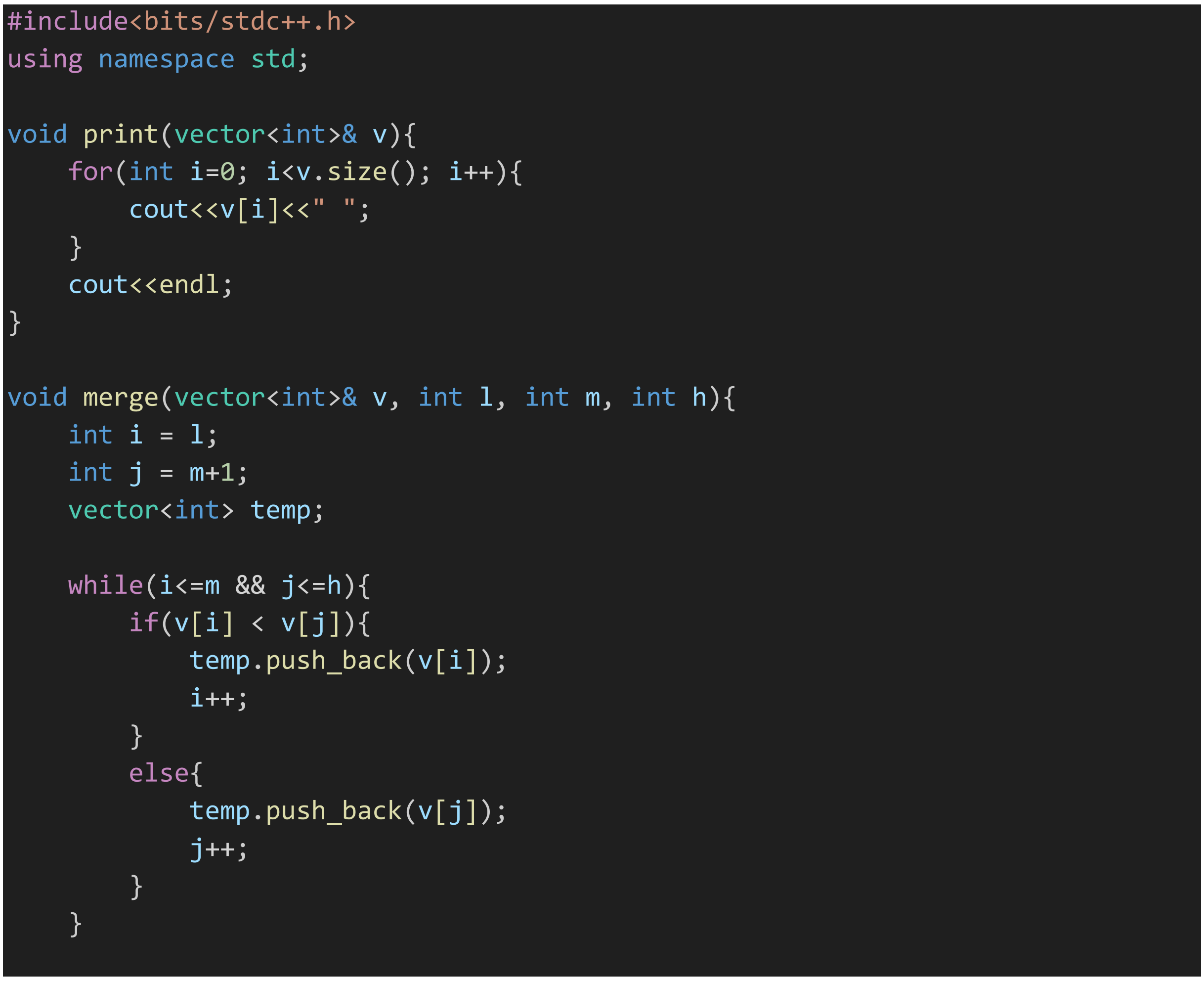


```cpp
#include<bits/stdc++.h>
using namespace std;

void print(vector<int>& v){
    for(int i=0; i<v.size(); i++){
        cout<<v[i]<<" ";
    }
    cout<<endl;
}

void selection(vector<int>& v){                 //TC = O(N^2) FOR ALL CASES
    for(int i=0; i<v.size()-1; i++){            //SC = O(1)
        int mini = i;
        for(int j=i+1; j<v.size(); j++){
            if(v[mini] > v[j]){
                mini = j;
            }
        }
        swap(v[i], v[mini]);
    }
}

void bubble(vector<int>& v){                   //TC = O(N^2) FOR WORST AND AVG
CASE
    for(int i=v.size()-1; i>=0; i--){          //TC = O(N) FOR BEST CASE
        bool swapped = false;                  //SC = O(1)
        for(int j=0; j<v.size()-1; j++){
            if(v[j] > v[j+1]){
                swap(v[j], v[j+1]);
                swapped = true;
            }
        }
        if(!swapped) break;
    }
}

void insertion(vector<int>& v){
    for(int i=0; i<v.size(); i++){
        int j = i;
        while(j>0){
            if(v[j-1] > v[j]){
                swap(v[j-1], v[j]);
                j--;
            }
            else break;
        }
    }
}
```



```cpp
int main(){

    // vector<int>v = {1,2,3,46,34,57,7,5,437,345,23,78};
    vector<int>v = {1,2,3,46,34,57,7,5,437,345,23,78};
    print(v);

    // selection(v);

    // bubble(v);

    // insertion(v);
    // print(v);

    return 0;
}
```

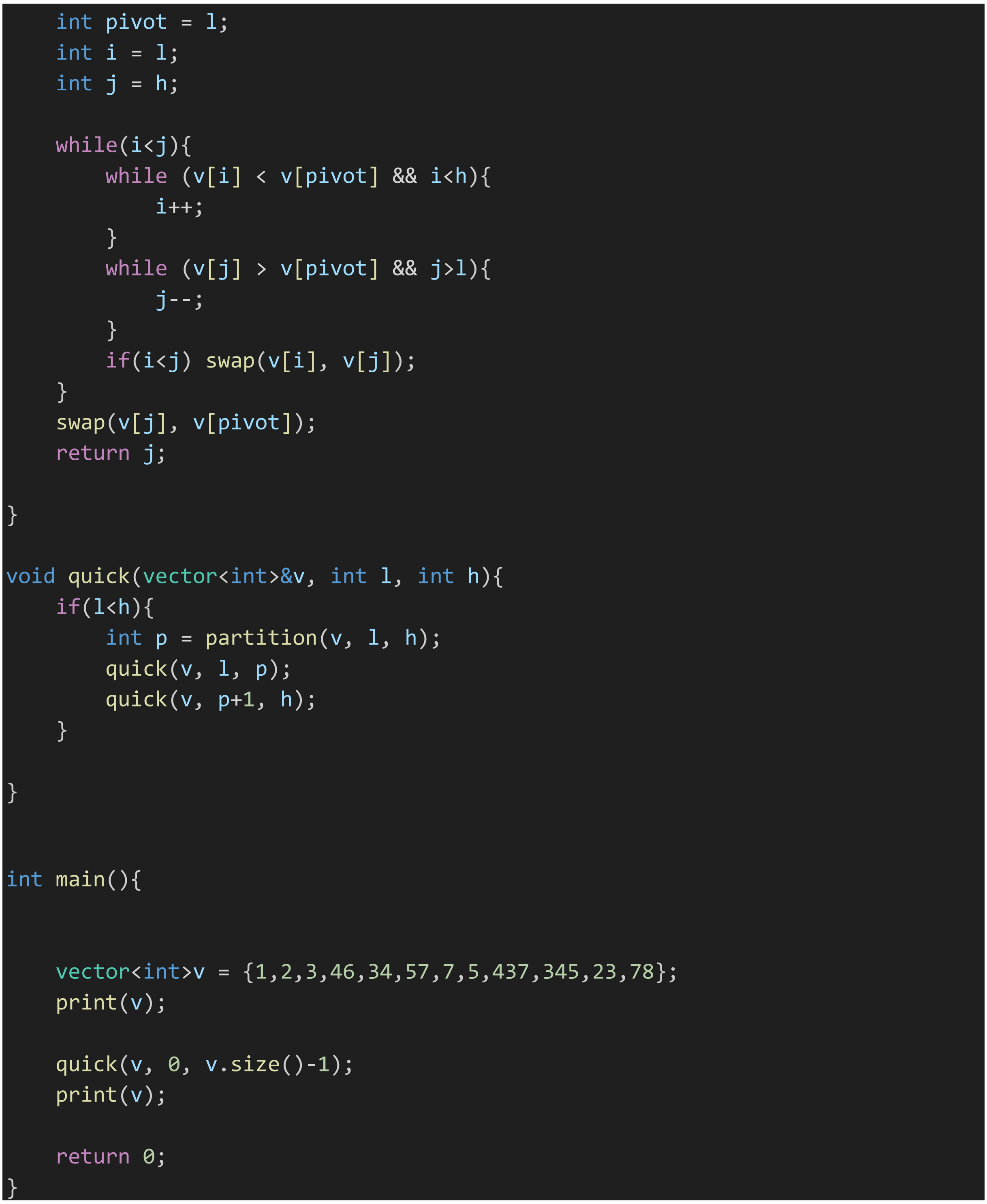
Merge:



```cpp
#include<bits/stdc++.h>
using namespace std;

void print(vector<int>& v){
    for(int i=0; i<v.size(); i++){
        cout<<v[i]<<" ";
    }
    cout<<endl;
}

void merge(vector<int>& v, int l, int m, int h){
    int i = l;
    int j = m+1;
    vector<int> temp;

    while(i<=m && j<=h){
        if(v[i] < v[j]){
            temp.push_back(v[i]);
            i++;
        }
        else{
            temp.push_back(v[j]);
            j++;
        }
    }
    while(i<=m){
        temp.push_back(v[i]);
        i++;
    }
    while(j<=h){
        temp.push_back(v[j]);
        j++;
    }

    for (int i = l; i <= h; i++)
    {
        v[i] = temp[i-l];
    }

}

void merge_sort(vector<int>& v,int l, int h){
    if(l >= h) return;
    int mid = (l+h)/2;
    merge_sort(v, l, mid);
    merge_sort(v, mid+1, h);
    merge(v, l, mid, h);
}


int main(){

    vector<int> v = {1,2,3,46,34,57,7,5,437,345,23,78};
    print(v);

    merge_sort(v, 0, v.size()-1);
    print(v);
    return 0;
}
```

Quick Sort:

```
#include<bits/stdc++.h>
using namespace std;

void print(vector<int> v){
    for(int i=0; i<v.size(); i++){
        cout<<v[i]<<" ";
    }
    cout<<endl;
}

int partition(vector<int>& v, int l, int h){
```



```
    int pivot = l;
    int i = l;
    int j = h;

    while(i<j){
        while (v[i] < v[pivot] && i<h){
            i++;
        }
        while (v[j] > v[pivot] && j>l){
            j--;
        }
        if(i<j) swap(v[i], v[j]);
    }
    swap(v[j], v[pivot]);
    return j;

}

void quick(vector<int>&v, int l, int h){
    if(l<h){
        int p = partition(v, l, h);
        quick(v, l, p);
        quick(v, p+1, h);
    }

}


int main(){


    vector<int>v = {1,2,3,46,34,57,7,5,437,345,23,78};
    print(v);

    quick(v, 0, v.size()-1);
    print(v);

    return 0;
}
```