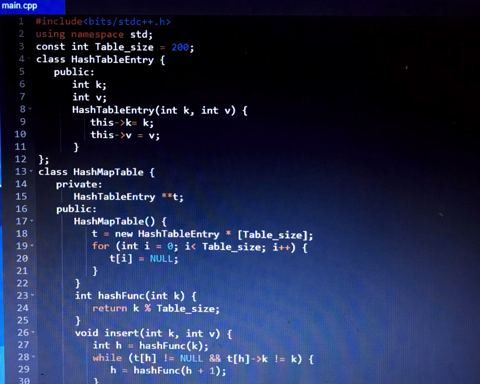
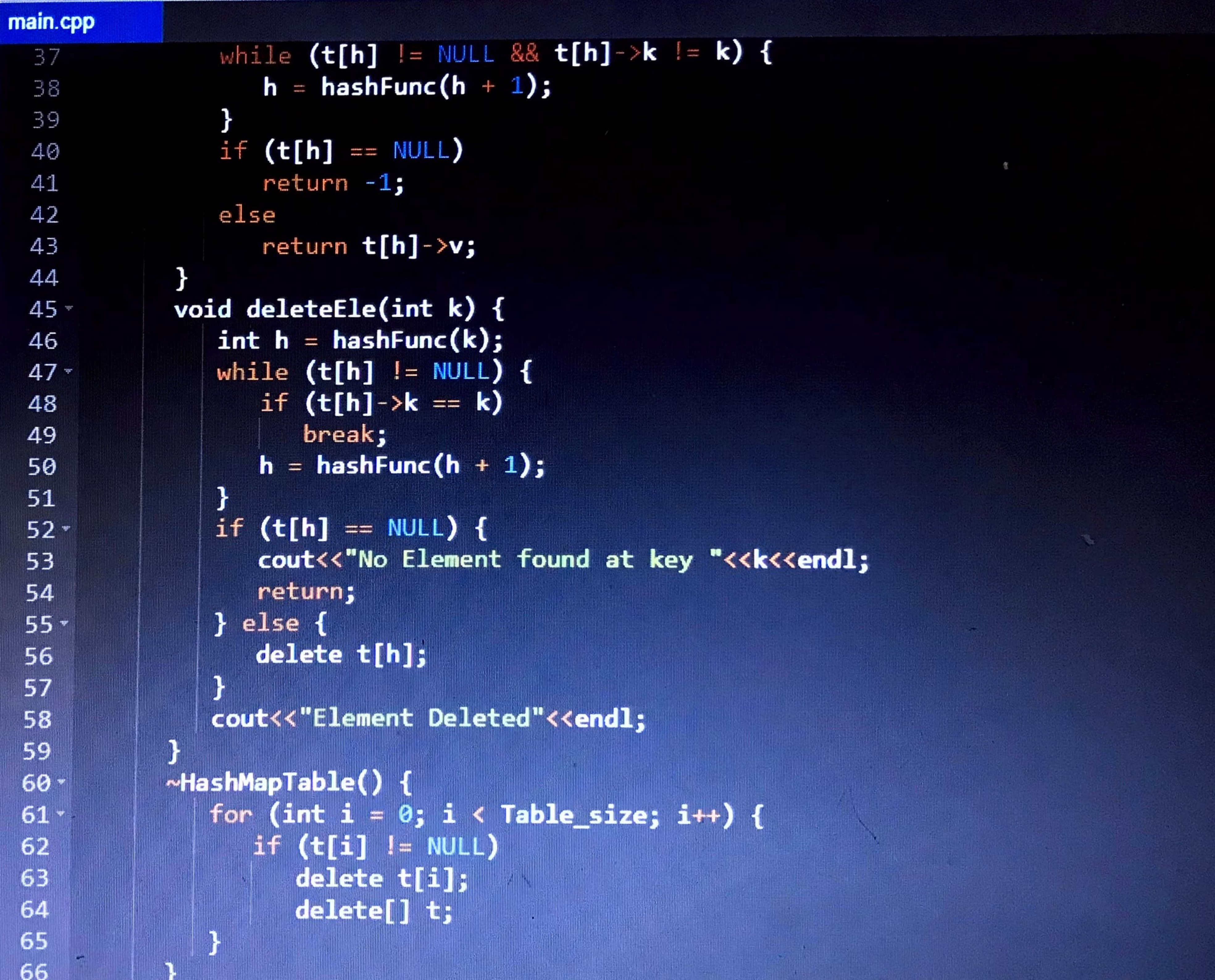
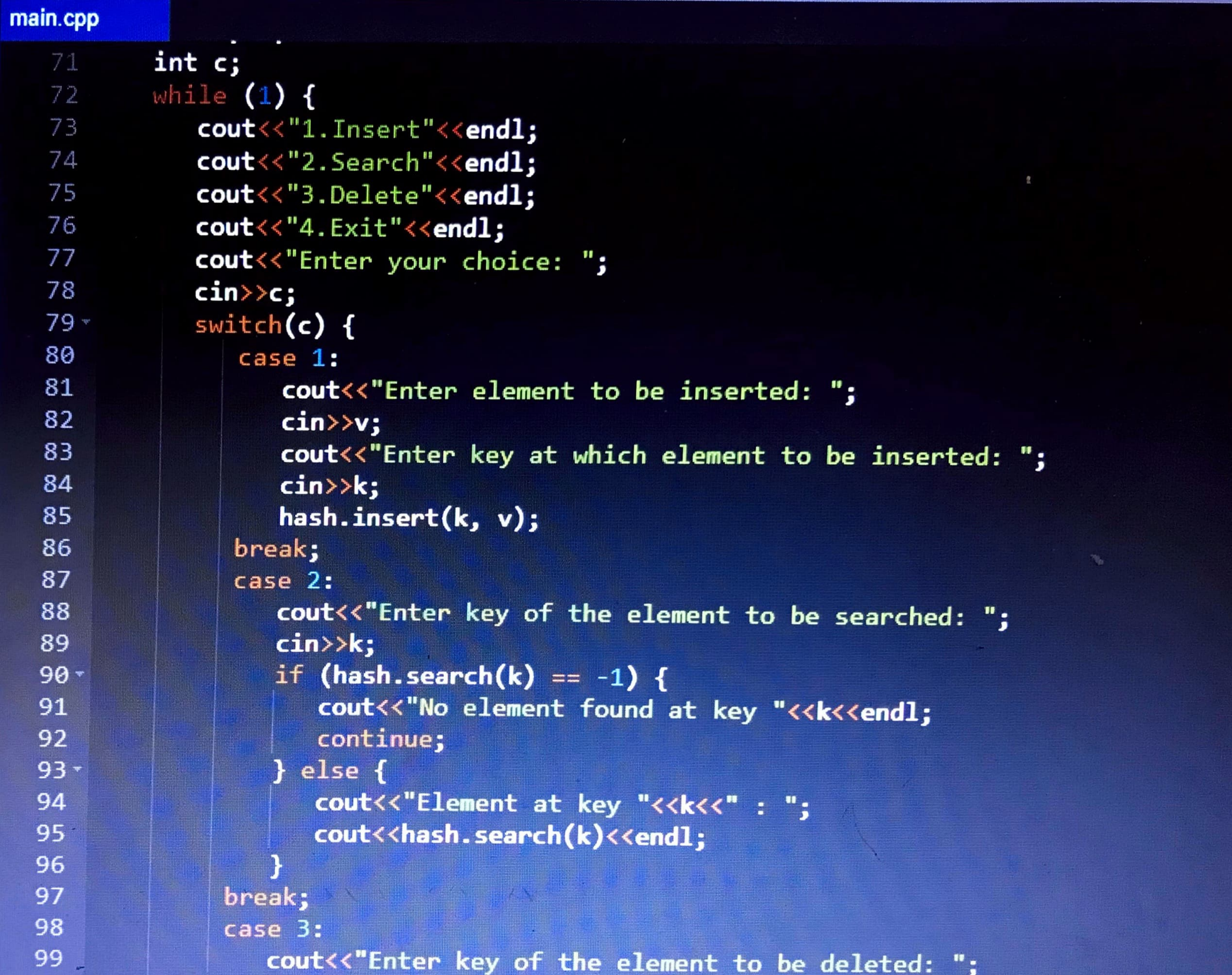


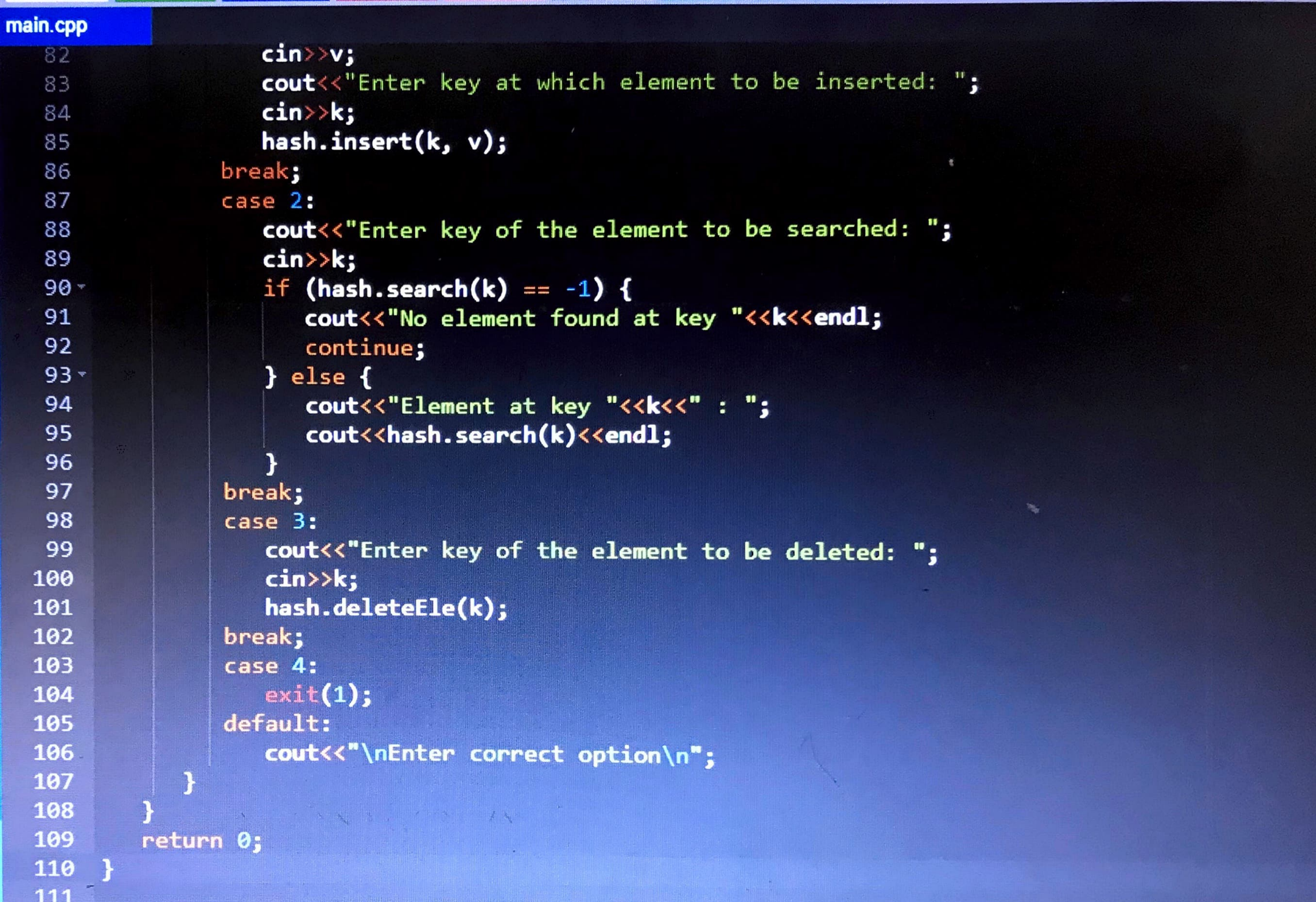
```
#include<bits/stdc++.h>
using namespace std;
const int Table_size = 200;
class HashTableEntry {
   public:
      int k;
      int v;
      HashTableEntry(int k, int v) {
         this->k= k;
         this->v = v;
      }
};
class HashMapTable {
   private:
      HashTableEntry **t;
   public:
      HashMapTable() {
         t = new HashTableEntry * [Table_size];
         for (int i = 0; i< Table_size; i++) {
            t[i] = NULL;
         }
      }
      int hashFunc(int k) {
         return k % Table_size;
      }
      void insert(int k, int v) {
         int h = hashFunc(k);
         while (t[h] != NULL && t[h]->k != k) {
            h = hashFunc(h + 1);
         }
```



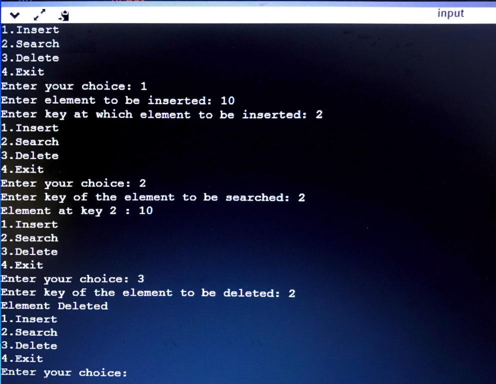
```cpp
        while (t[h] != NULL && t[h]->k != k) {
            h = hashFunc(h + 1);
        }
        if (t[h] == NULL)
            return -1;
        else
            return t[h]->v;
    }
    void deleteEle(int k) {
        int h = hashFunc(k);
        while (t[h] != NULL) {
            if (t[h]->k == k)
                break;
            h = hashFunc(h + 1);
        }
        if (t[h] == NULL) {
            cout<<"No Element found at key "<<k<<endl;
            return;
        } else {
            delete t[h];
        }
        cout<<"Element Deleted"<<endl;
    }
    ~HashMapTable() {
        for (int i = 0; i < Table_size; i++) {
            if (t[i] != NULL)
                delete t[i];
                delete[] t;
        }
    }
```



```cpp
    int c;
    while (1) {
        cout<<"1.Insert"<<endl;
        cout<<"2.Search"<<endl;
        cout<<"3.Delete"<<endl;
        cout<<"4.Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>c;
        switch(c) {
            case 1:
                cout<<"Enter element to be inserted: ";
                cin>>v;
                cout<<"Enter key at which element to be inserted: ";
                cin>>k;
                hash.insert(k, v);
            break;
            case 2:
                cout<<"Enter key of the element to be searched: ";
                cin>>k;
                if (hash.search(k) == -1) {
                    cout<<"No element found at key "<<k<<endl;
                    continue;
                } else {
                    cout<<"Element at key "<<k<<" : ";
                    cout<<hash.search(k)<<endl;
                }
            break;
            case 3:
                cout<<"Enter key of the element to be deleted: ";
```



```cpp
                cin>>v;
                cout<<"Enter key at which element to be inserted: ";
                cin>>k;
                hash.insert(k, v);
            break;
            case 2:
                cout<<"Enter key of the element to be searched: ";
                cin>>k;
                if (hash.search(k) == -1) {
                    cout<<"No element found at key "<<k<<endl;
                    continue;
                } else {
                    cout<<"Element at key "<<k<<" : ";
                    cout<<hash.search(k)<<endl;
                }
            break;
            case 3:
                cout<<"Enter key of the element to be deleted: ";
                cin>>k;
                hash.deleteEle(k);
            break;
            case 4:
                exit(1);
            default:
                cout<<"\nEnter correct option\n";
        }
    }
    return 0;
}
```



```
1.Insert
2.Search
3.Delete
4.Exit
Enter your choice: 1
Enter element to be inserted: 10
Enter key at which element to be inserted: 2
1.Insert
2.Search
3.Delete
4.Exit
Enter your choice: 2
Enter key of the element to be searched: 2
Element at key 2 : 10
1.Insert
2.Search
3.Delete
4.Exit
Enter your choice: 3
Enter key of the element to be deleted: 2
Element Deleted
1.Insert
2.Search
3.Delete
4.Exit
Enter your choice:
```