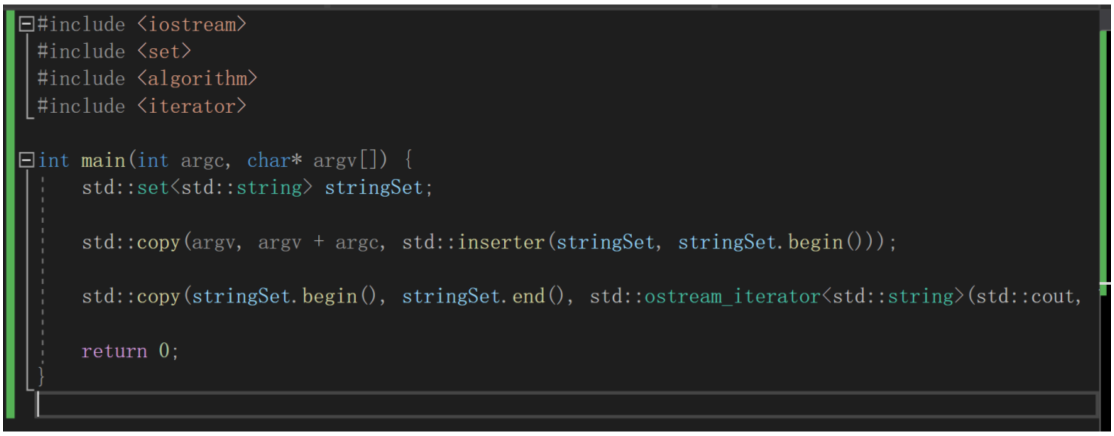


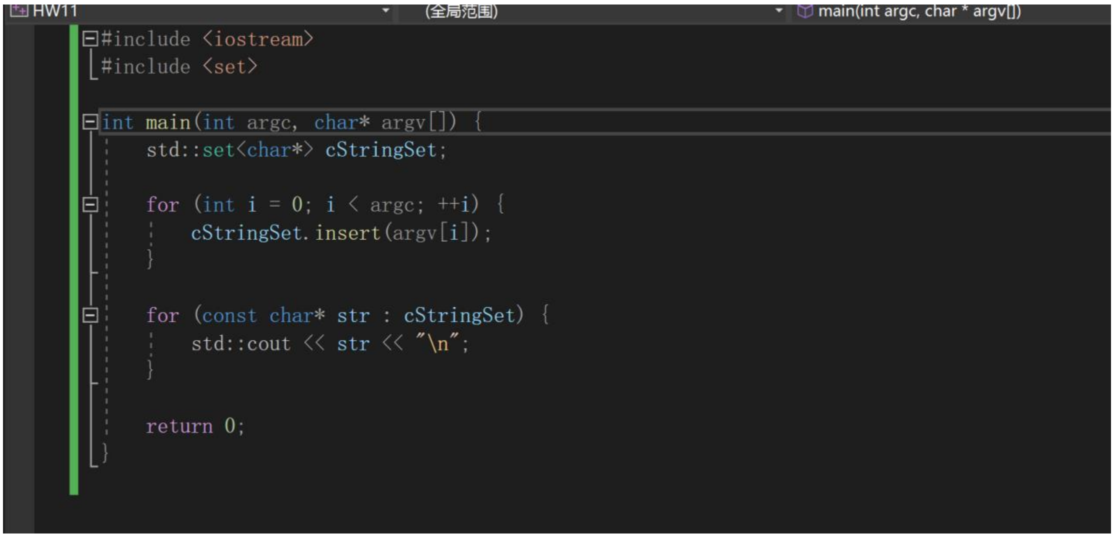
```
#include <iostream>
#include <set>
#include <algorithm>
#include <iterator>

int main(int argc, char* argv[]) {
    std::set<std::string> stringSet;

    std::copy(argv, argv + argc, std::inserter(stringSet, stringSet.begin()));

    std::copy(stringSet.begin(), stringSet.end(), std::ostream_iterator<std::string>(std::cout,

    return 0;
}
```



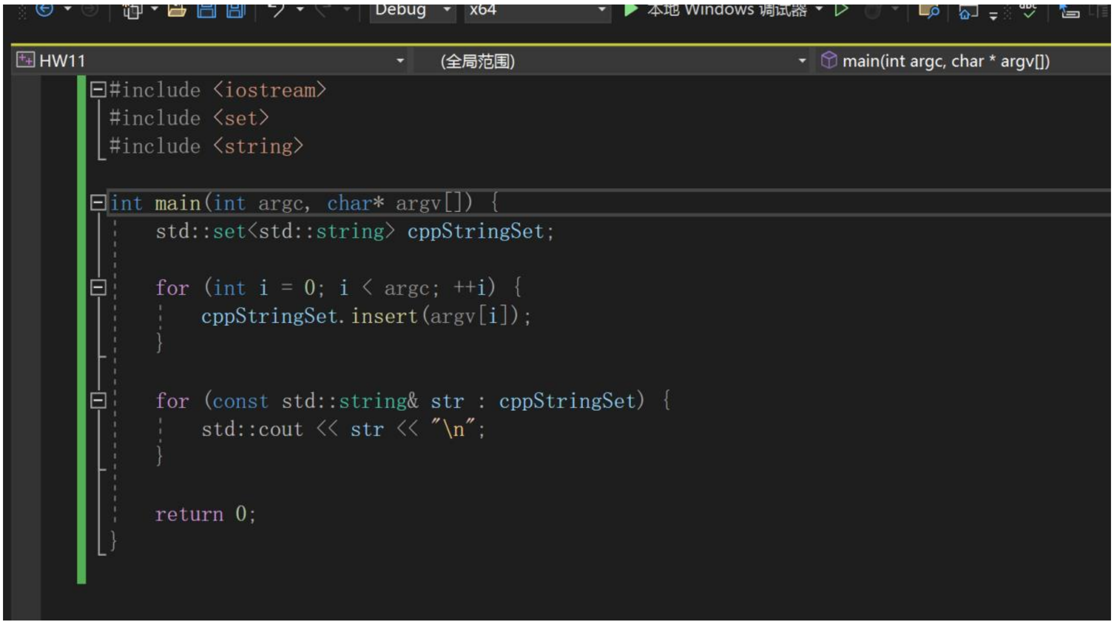
```
#include <iostream>
#include <set>

int main(int argc, char* argv[]) {
    std::set<char*> cStringSet;

    for (int i = 0; i < argc; ++i) {
        cStringSet.insert(argv[i]);
    }

    for (const char* str : cStringSet) {
        std::cout << str << "\n";
    }

    return 0;
}
```



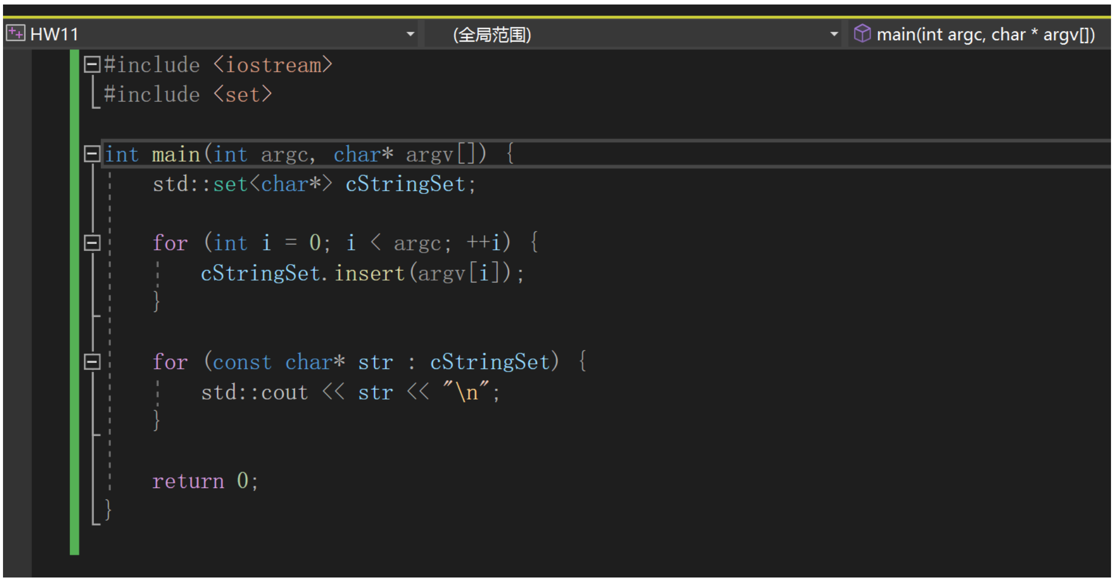
```
#include <iostream>
#include <set>
#include <string>

int main(int argc, char* argv[]) {
    std::set<std::string> cppStringSet;

    for (int i = 0; i < argc; ++i) {
        cppStringSet.insert(argv[i]);
    }

    for (const std::string& str : cppStringSet) {
        std::cout << str << "\n";
    }

    return 0;
}
```



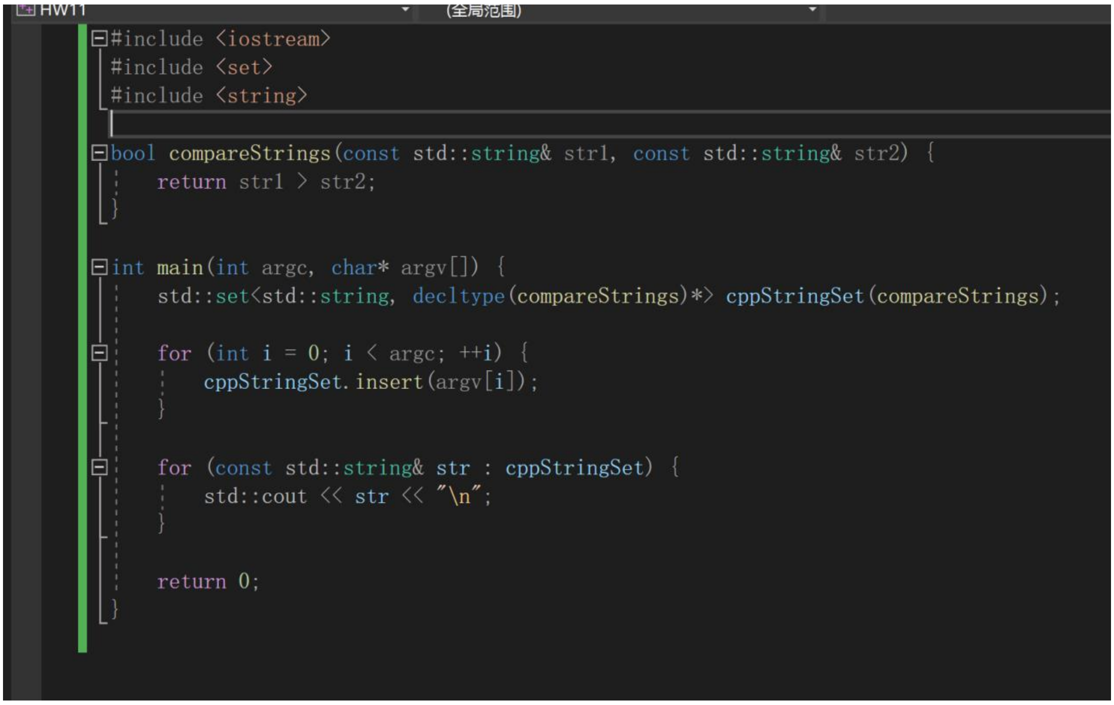
```cpp
#include <iostream>
#include <set>

int main(int argc, char* argv[]) {
    std::set<char*> cStringSet;

    for (int i = 0; i < argc; ++i) {
        cStringSet.insert(argv[i]);
    }

    for (const char* str : cStringSet) {
        std::cout << str << "\n";
    }

    return 0;
}
```



```cpp
#include <iostream>
#include <set>
#include <string>

bool compareStrings(const std::string& str1, const std::string& str2) {
    return str1 > str2;
}

int main(int argc, char* argv[]) {
    std::set<std::string, decltype(compareStrings)*> cppStringSet(compareStrings);

    for (int i = 0; i < argc; ++i) {
        cppStringSet.insert(argv[i]);
    }

    for (const std::string& str : cppStringSet) {
        std::cout << str << "\n";
    }

    return 0;
}
```