Lab Report

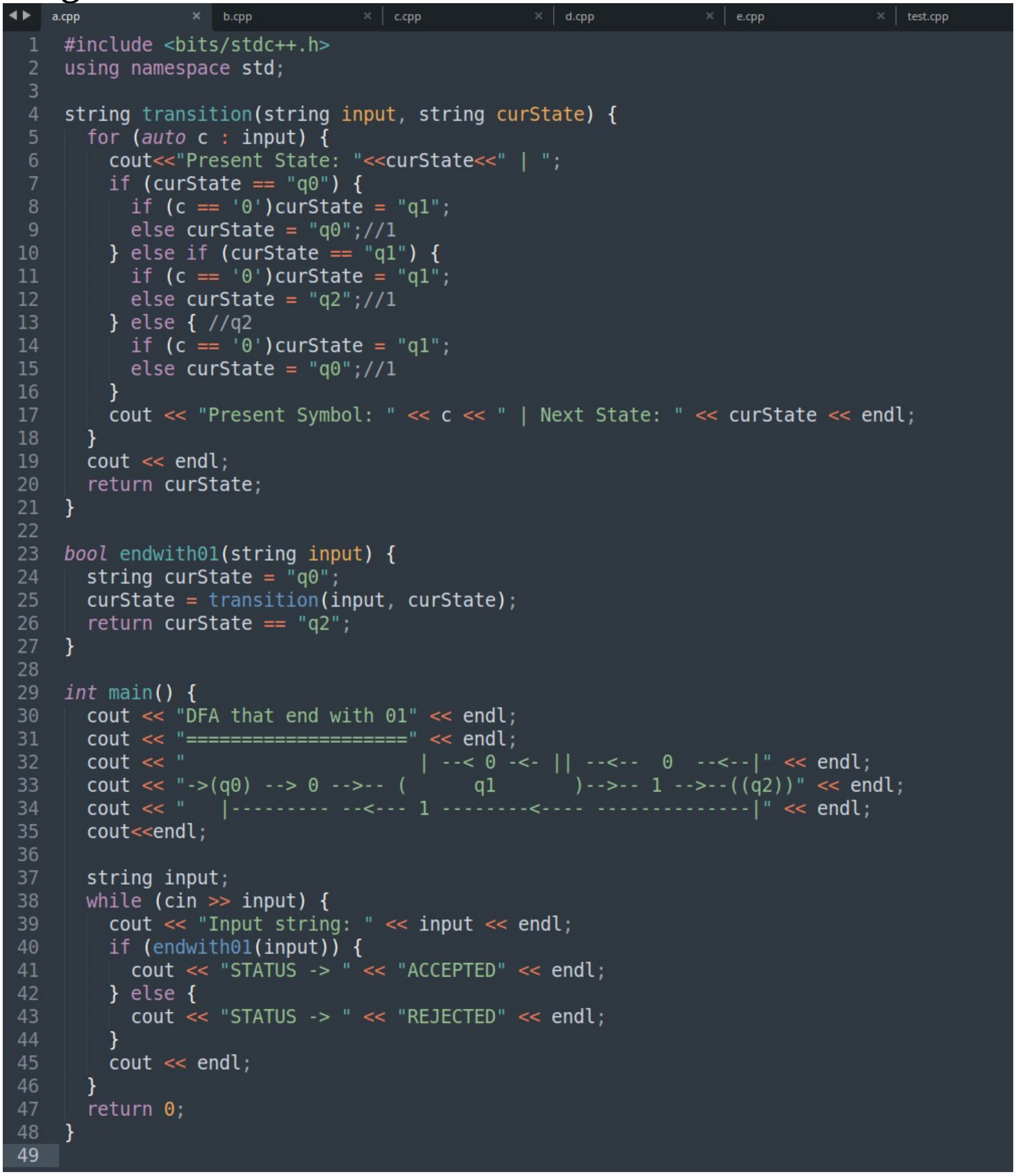
Subject: CSE-3214 Compiler Design Lab
Experiment Name: Design and Implementation of DFA and NFA that end with “01”
Tool Used: C++ Programming Language

Submitted By:
Eyasir Ahamed
Roll No:15
Registration No: 202004017

Submitted To:
Kohinur Parvin
Lecturer,
Netrokona University, Netrokona

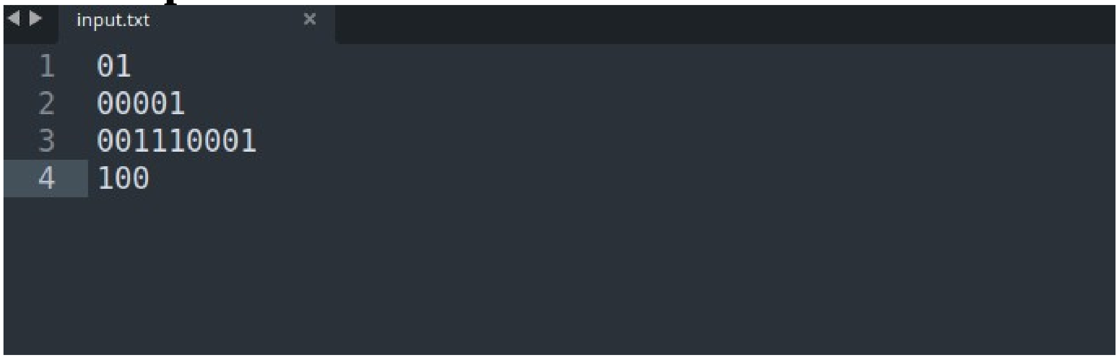
DFA That End with “01”

Program Code:



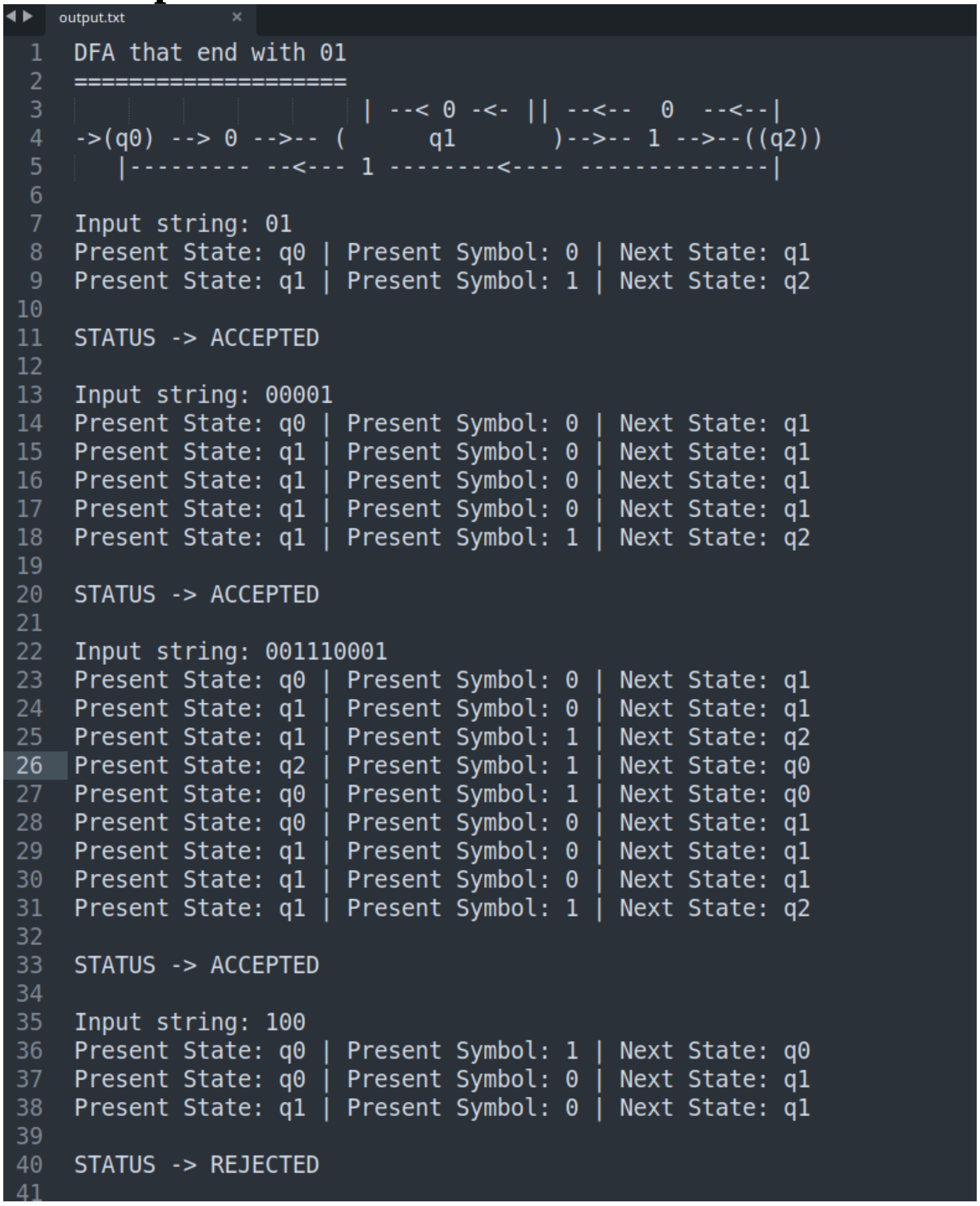
```cpp
#include <bits/stdc++.h>
using namespace std;

string transition(string input, string curState) {
  for (auto c : input) {
    cout<<"Present State: "<<curState<<" | ";
    if (curState == "q0") {
      if (c == '0')curState = "q1";
      else curState = "q0";//1
    } else if (curState == "q1") {
      if (c == '0')curState = "q1";
      else curState = "q2";//1
    } else { //q2
      if (c == '0')curState = "q1";
      else curState = "q0";//1
    }
    cout << "Present Symbol: " << c << " | Next State: " << curState << endl;
  }
  cout << endl;
  return curState;
}

bool endwith01(string input) {
  string curState = "q0";
  curState = transition(input, curState);
  return curState == "q2";
}

int main() {
  cout << "DFA that end with 01" << endl;
  cout << "====================" << endl;
  cout << "                     | --< 0 -<- || --<--  0  --<--|" << endl;
  cout << "->(q0) --> 0 -->-- (      q1       )-->-- 1 -->--((q2))" << endl;
  cout << "   |--------- --<--- 1 --------<---- --------------|" << endl;
  cout<<endl;

  string input;
  while (cin >> input) {
    cout << "Input string: " << input << endl;
    if (endwith01(input)) {
      cout << "STATUS -> " << "ACCEPTED" << endl;
    } else {
      cout << "STATUS -> " << "REJECTED" << endl;
    }
    cout << endl;
  }
  return 0;
}
```

Input Example:



```
01
00001
001110001
100
```

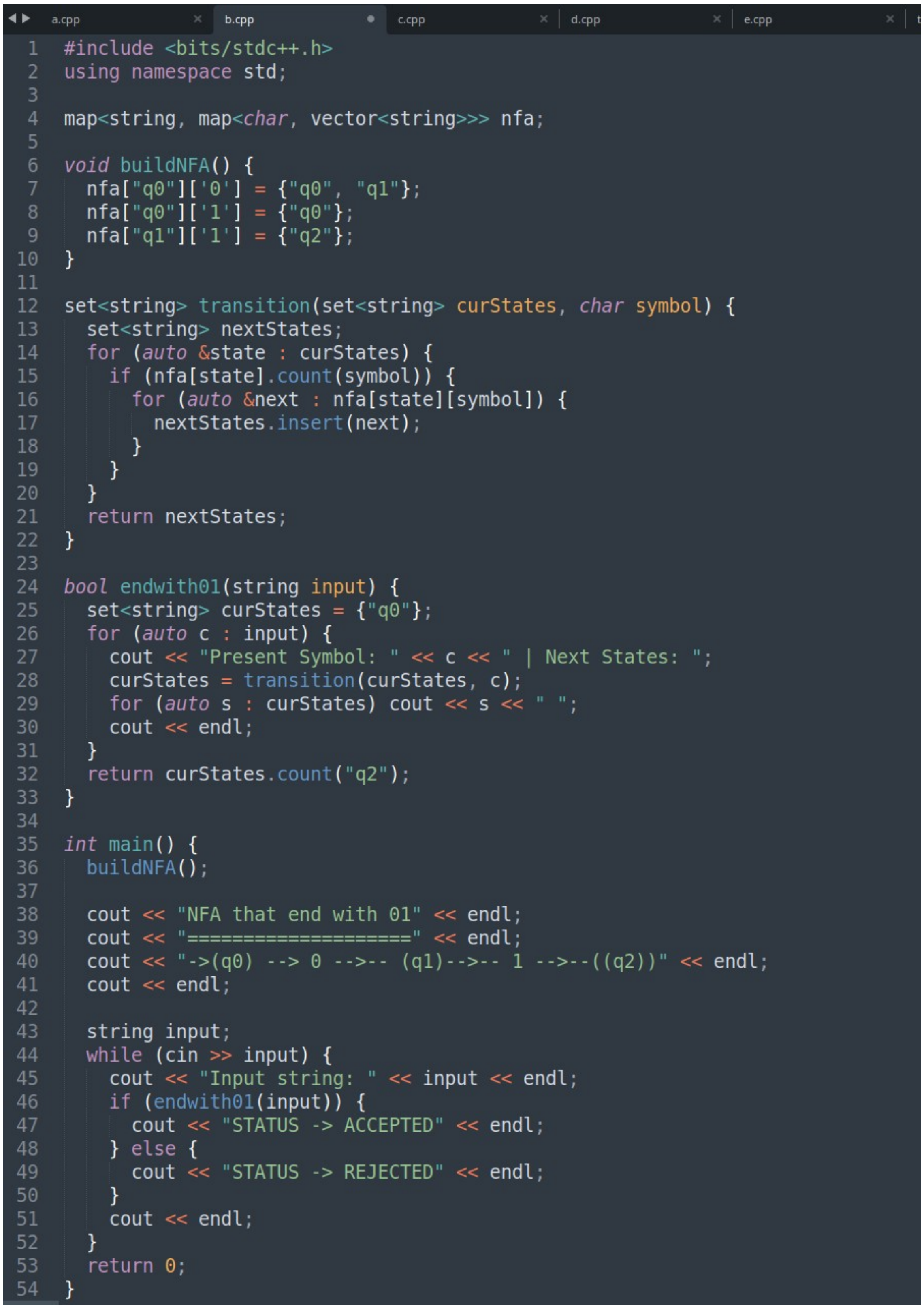
Output Example:



```
DFA that end with 01
====================
                     | --< 0 -<- || --<--  0  --<--|
->(q0) --> 0 -->-- (      q1        )-->-- 1 -->--((q2))
   |--------- --<--- 1 --------<---- --------------|

Input string: 01
Present State: q0 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 1 | Next State: q2

STATUS -> ACCEPTED

Input string: 00001
Present State: q0 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 1 | Next State: q2

STATUS -> ACCEPTED

Input string: 001110001
Present State: q0 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 1 | Next State: q2
Present State: q2 | Present Symbol: 1 | Next State: q0
Present State: q0 | Present Symbol: 1 | Next State: q0
Present State: q0 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 1 | Next State: q2

STATUS -> ACCEPTED

Input string: 100
Present State: q0 | Present Symbol: 1 | Next State: q0
Present State: q0 | Present Symbol: 0 | Next State: q1
Present State: q1 | Present Symbol: 0 | Next State: q1

STATUS -> REJECTED
```

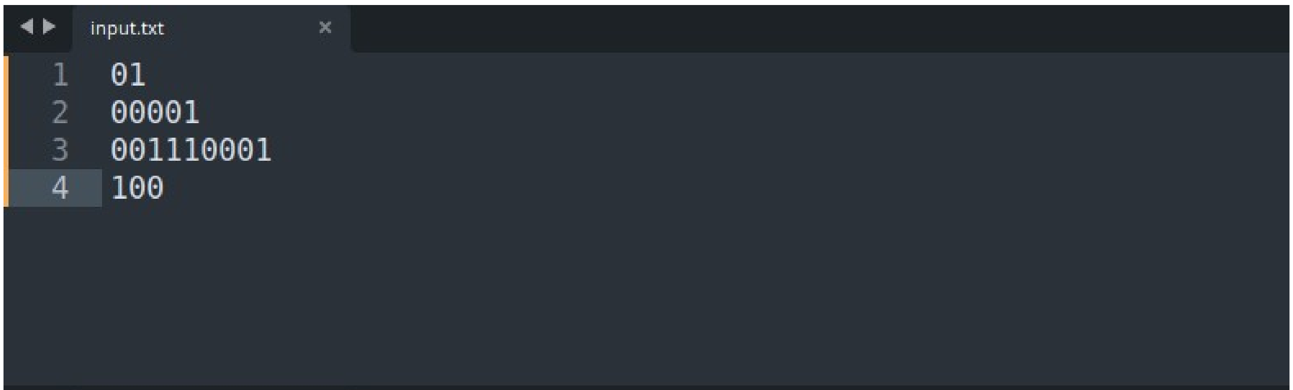
NFA that end with “01:

<u>Program Code</u>



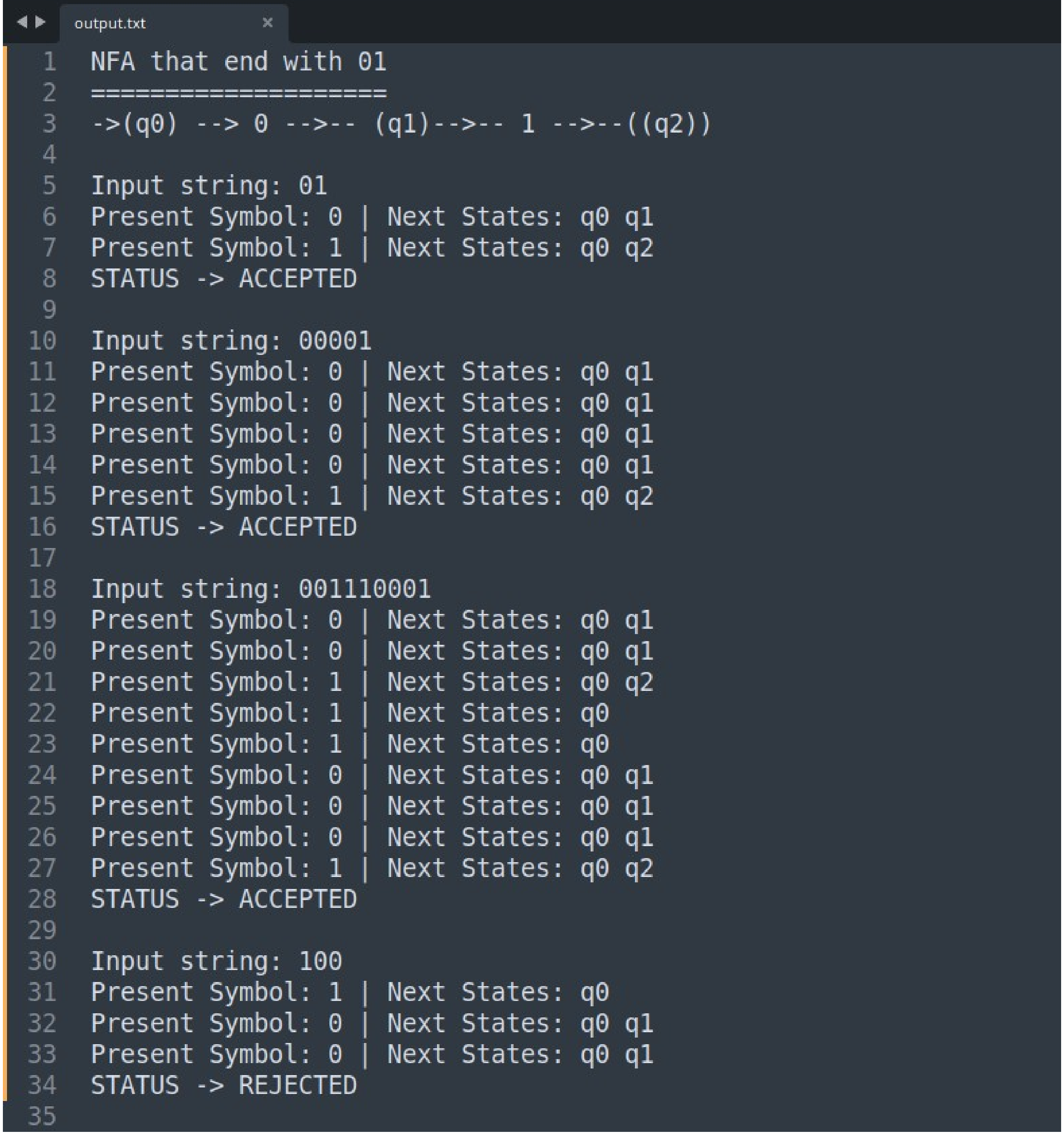
```cpp
#include <bits/stdc++.h>
using namespace std;

map<string, map<char, vector<string>>> nfa;

void buildNFA() {
  nfa["q0"]['0'] = {"q0", "q1"};
  nfa["q0"]['1'] = {"q0"};
  nfa["q1"]['1'] = {"q2"};
}

set<string> transition(set<string> curStates, char symbol) {
  set<string> nextStates;
  for (auto &state : curStates) {
    if (nfa[state].count(symbol)) {
      for (auto &next : nfa[state][symbol]) {
        nextStates.insert(next);
      }
    }
  }
  return nextStates;
}

bool endwith01(string input) {
  set<string> curStates = {"q0"};
  for (auto c : input) {
    cout << "Present Symbol: " << c << " | Next States: ";
    curStates = transition(curStates, c);
    for (auto s : curStates) cout << s << " ";
    cout << endl;
  }
  return curStates.count("q2");
}

int main() {
  buildNFA();

  cout << "NFA that end with 01" << endl;
  cout << "====================" << endl;
  cout << "->(q0) --> 0 -->-- (q1)-->-- 1 -->--((q2))" << endl;
  cout << endl;

  string input;
  while (cin >> input) {
    cout << "Input string: " << input << endl;
    if (endwith01(input)) {
      cout << "STATUS -> ACCEPTED" << endl;
    } else {
      cout << "STATUS -> REJECTED" << endl;
    }
    cout << endl;
  }
  return 0;
}
```

Input and Output Example



input.txt

```
01
00001
001110001
100
```



output.txt

```
NFA that end with 01
====================
->(q0) --> 0 -->-- (q1)-->-- 1 -->--((q2))

Input string: 01
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 1 | Next States: q0 q2
STATUS -> ACCEPTED

Input string: 00001
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 1 | Next States: q0 q2
STATUS -> ACCEPTED

Input string: 001110001
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 1 | Next States: q0 q2
Present Symbol: 1 | Next States: q0
Present Symbol: 1 | Next States: q0
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 1 | Next States: q0 q2
STATUS -> ACCEPTED

Input string: 100
Present Symbol: 1 | Next States: q0
Present Symbol: 0 | Next States: q0 q1
Present Symbol: 0 | Next States: q0 q1
STATUS -> REJECTED

```