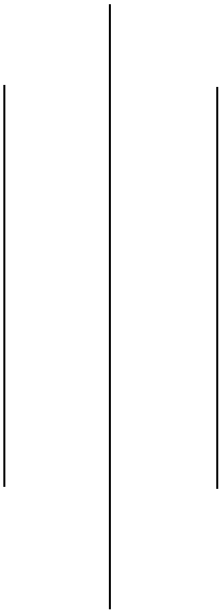
National College of Computer Studies

Paknajol, Kathmandu



Lab Report 4

Submitted by:

Atullya Maharjan

BSc. CSIT 4th Semester

Roll. No: 5

Submitted to:

Prashant Gautam

NCCS

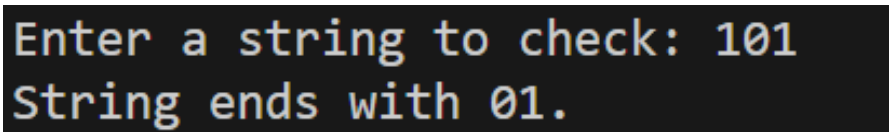
1. WAP to design a NFA that accepts string ending with 01.

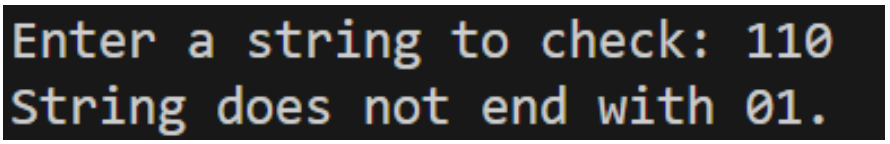
```
#include <iostream>
#include <vector>
using namespace std;
vector<int> states = {0, 1, 2};
vector<vector<pair<char, int>>> transitions = {
    {{'0', 0}, {'1', 0}, {'0', 1}},
    {{'1', 2}},
    {{}}};
bool simulate_nfa(string input){
    vector<int> current_states = {0};
    for (char c : input){

        vector<int> next_states;
        for (int state : current_states)
        {
            for (auto transition : transitions[state])
            {
                if (transition.first == c)
                {
                    next_states.push_back(transition.second);
```

```
            }
        }
    }

    if (next_states.empty())
    {
        return false;
    }

    current_states = next_states;
}

for (int state : current_states)
{
    if (state == 2)
    {
        return true;
    }
}

return false;
}


int main()
{
    string input;
    cout << "Enter a string to check: ";
    cin >> input;
```

```
    if (simulate_nfa(input))
    {
        cout << "String ends with 01." << endl;
    }
    else
    {
        cout << "String does not end with 01." << endl;
    }
    return 0;
}
```

OUTPUT





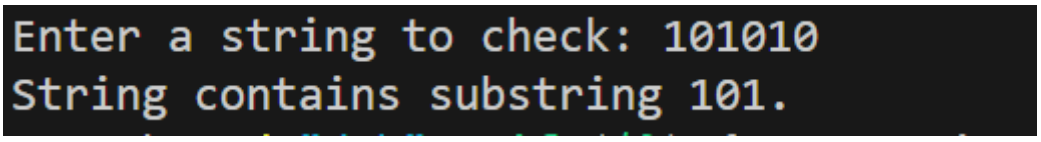
2. WAP to design a NFA that accepts string containing substring 101

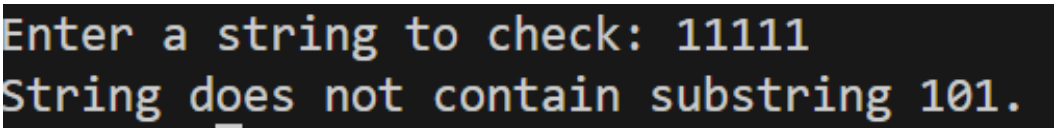
```
#include <iostream>
#include <vector>

using namespace std;


vector<int> states = {0, 1, 2, 3};
vector<vector<pair<char, int>>> transitions = {
    {{'0', 0}, {'1', 0}, {'1', 1}},
    {{'0', 2}},
    {{'1', 3}},
    {{'0', 3}, {'1', 3}}};

bool simulate_nfa(string input)
{
    vector<int> current_states = {0};

    for (char c : input)
    {
        vector<int> next_states;
        for (int state : current_states)
        {
            for (auto transition : transitions[state])
            {
```

```
            if (transition.first == c)
            {
                next_states.push_back(transition.second);
            }
        }
    }
    if (next_states.empty())
    {
        return false;
    }
    current_states = next_states;
  }
  for (int state : current_states)
  {
    if (state == 3)
    {
        return true;
    }
  }
  return false;
}

int main()
{
```

```
    string input;
    cout << "Enter a string to check: ";
    cin >> input;

    if (simulate_nfa(input))
    {
        cout << "String contains substring 101." << endl;
    }
    else
    {
        cout << "String does not contain substring 101." << endl;
    }

    return 0;
}
```

OUTPUT





3. WAP to design a NFA that accepts string starting with 10

```
#include <iostream>
#include <vector>

using namespace std;

vector<int> states = {0, 1, 2};
vector<vector<pair<char, int>>> transitions = {
    {{'1', 1}},
    {{'0', 2}},
    {{'0', 2}, {'1', 2}},
    {{}}};

bool simulate_nfa(string input)
{
    vector<int> current_states = {0};

    for (char c : input)
    {
        vector<int> next_states;
        for (int state : current_states)
        {
            for (auto transition : transitions[state])
            {
```

```
            if (transition.first == c)
            {
                next_states.push_back(transition.second);
            }
        }
    }
    if (next_states.empty())
    {
        return false;
    }
    current_states = next_states;
  }
  for (int state : current_states)
  {
    if (state == 2)
    {
        return true;
    }
  }
  return false;
}
int main()
{
```

```
    string input;
    cout << "Enter a string to check: ";
    cin >> input;

    if (simulate_nfa(input))
    {
        cout << "String starts with 10." << endl;
    }
    else
    {
        cout << "String does not start with 10." << endl;
    }

    return 0;
}
```

OUTPUT