Quinn Roemer

CISP - 440

Assignment 13.42

12/13/2018

Part 0 - State Machine Generator

Description:

The goal of this assignment was to implement some code that is able to generate a state table given a string. This code is then able to scan a set of characters until it finds a certain combination. It was our job to implement the function that would turn a string into a state table.

Source Code:

```
//Code written by Quinn Roemer, based on code by Professor Ross.
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

#define STATES 20

//Char array holds state table.
char nextState[STATES][2];

//Global variables to hold a count and the current state.
char state = 0;
int counter = 0;

//Hard coded input string
string instr = "abbbbbbaaaaaba";

//Prints out the nextState char.
void print_StateTable()
{
    cout << "s\ta\tb\n\n";

    int l = instr.length() + 1;
    for (int s = 0; s < l; s++) {
        cout << s << "\t";
        for (int ch = 0; ch < 2; ch++)
               cout << (int)nextState[s][ch] << "\t";
        cout << endl;
    }
}

//Compares first n characters of str1, with last n characters of str2.
int strncmp_olap(string str1, string str2, int n)
{
    int i, j;
```

```
    int l1 = str1.length();
    int l2 = str2.length();

    for (i = 0, j = l2 - n; i < n; i++, j++)
    {
        if (str1[i] != str2[j])
        {
                return true;
        }
    }

    return false;
}

//Generate a state table for a sequence detector
void generate_StateTable(void)
{
    int length = instr.length();

    string str;
    string current;

    for (int count = 0; count <= length; count++)
    {

        //Checking overlap on A.
        str.assign(instr, 0, count);
        current = str + 'a';

        cout << "State: " << count << endl;

        for (int index = current.length(); index >= 0; index--)
        {
                if (!strncmp_olap(instr, current, index))
                {

                        cout << "In state " << count << " go to state " << index << " for A" <<
endl;


                        //Place correct state in state table.
                        nextState[count][0] = index;

                        break;
                }
        }

        //Checking overlap on b.
        str.assign(instr, 0, count);
        current = str + 'b';
```

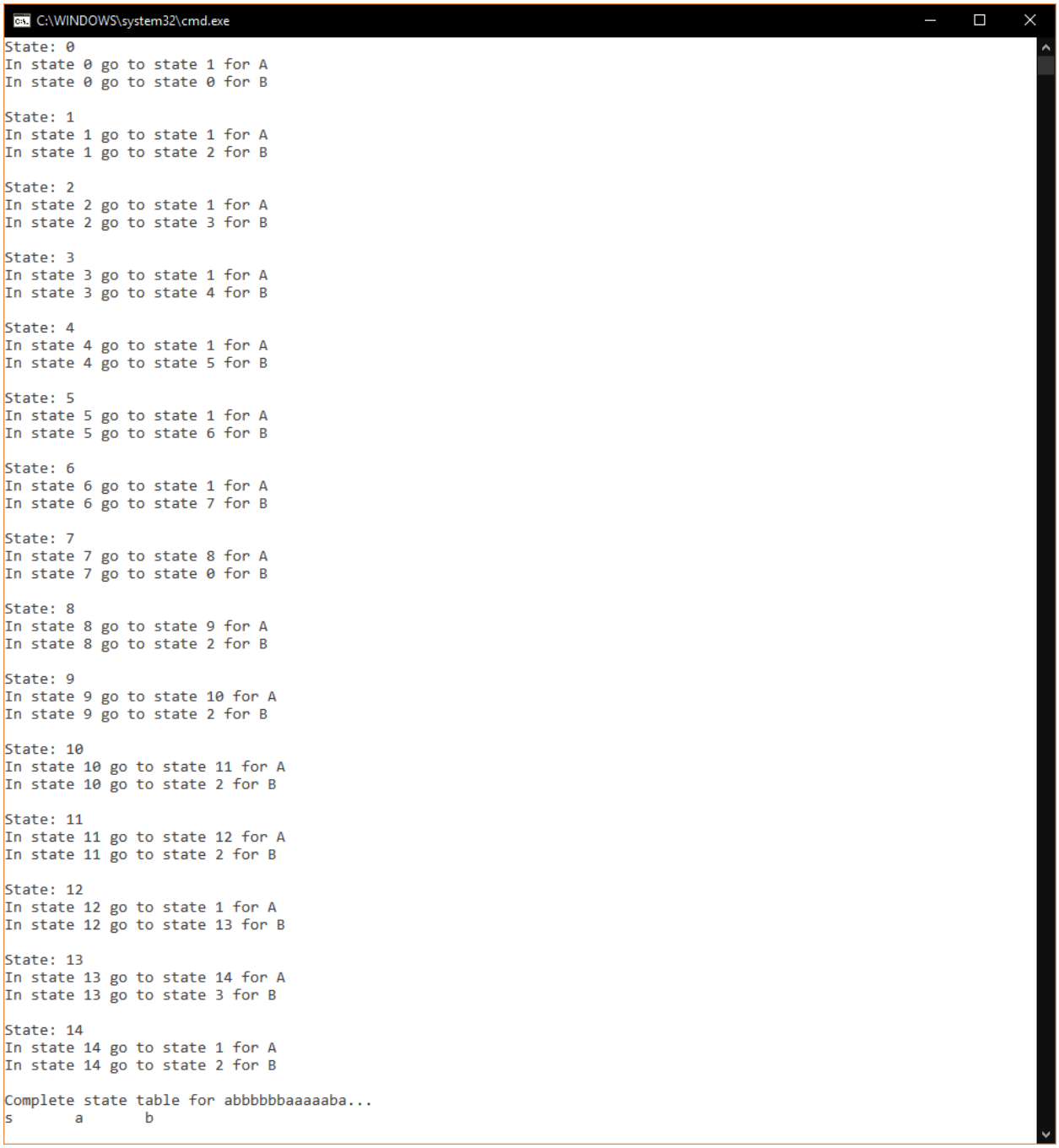
```
        for (int index = current.length(); index >= 0; index--)
        {
                if (!strncmp_olap(instr, current, index))
                {

                        cout << "In state " << count << " go to state " << index << " for B" <<
endl << endl;

                        //Place correct state in state table.
                        nextState[count][1] = index;
                        break;
                }
        }
    }
}

//A State Machine
void process(char ch)
{
    //Print current char
    //cout << ch;

    //Move to next state
    state = nextState[state][ch - 'a'];

    //Output count if we are in the accept state
    if (state == 14)
    {
        cout << counter + 2 << ", "; //+2 for notepad.
    }

    counter++;
}

//Main function to execute.
void main()
{
    generate_StateTable();
    cout << "Complete state table for " << instr << "..." << endl;
    print_StateTable();

    char ch;

    //Open a file
    ifstream in("monkeyData.txt");
    if (!in)
    {
        cout << "Error opening file";
        return;
    }
```

```
    cout << "\nThe string " << instr << " was found at locations: ";

    //Read characters from file and process them
    while (in)
    {
        in.get(ch);
        if (in)
        {
                process(ch);
        }
    }

    cout << endl;
}
```

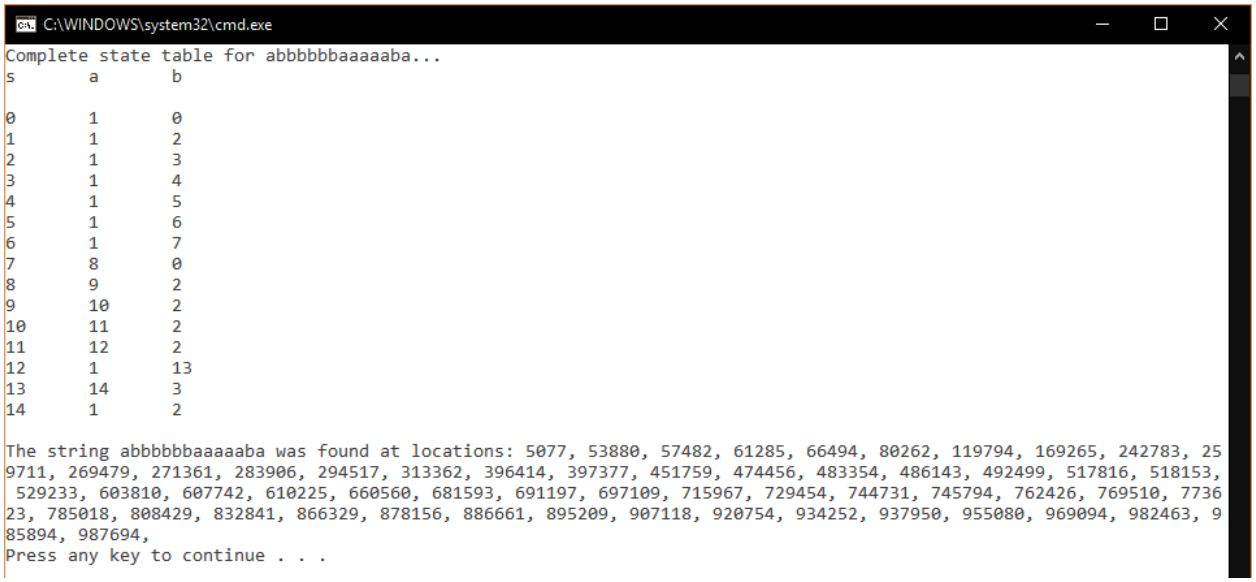
Output:

Note: The following 5 outputs will be for a given string in monkeyData.txt. Each output will be split into two pictures. The string it is for will be listed above the first picture of a given output.

String = "abbbbbbaaaaaba"

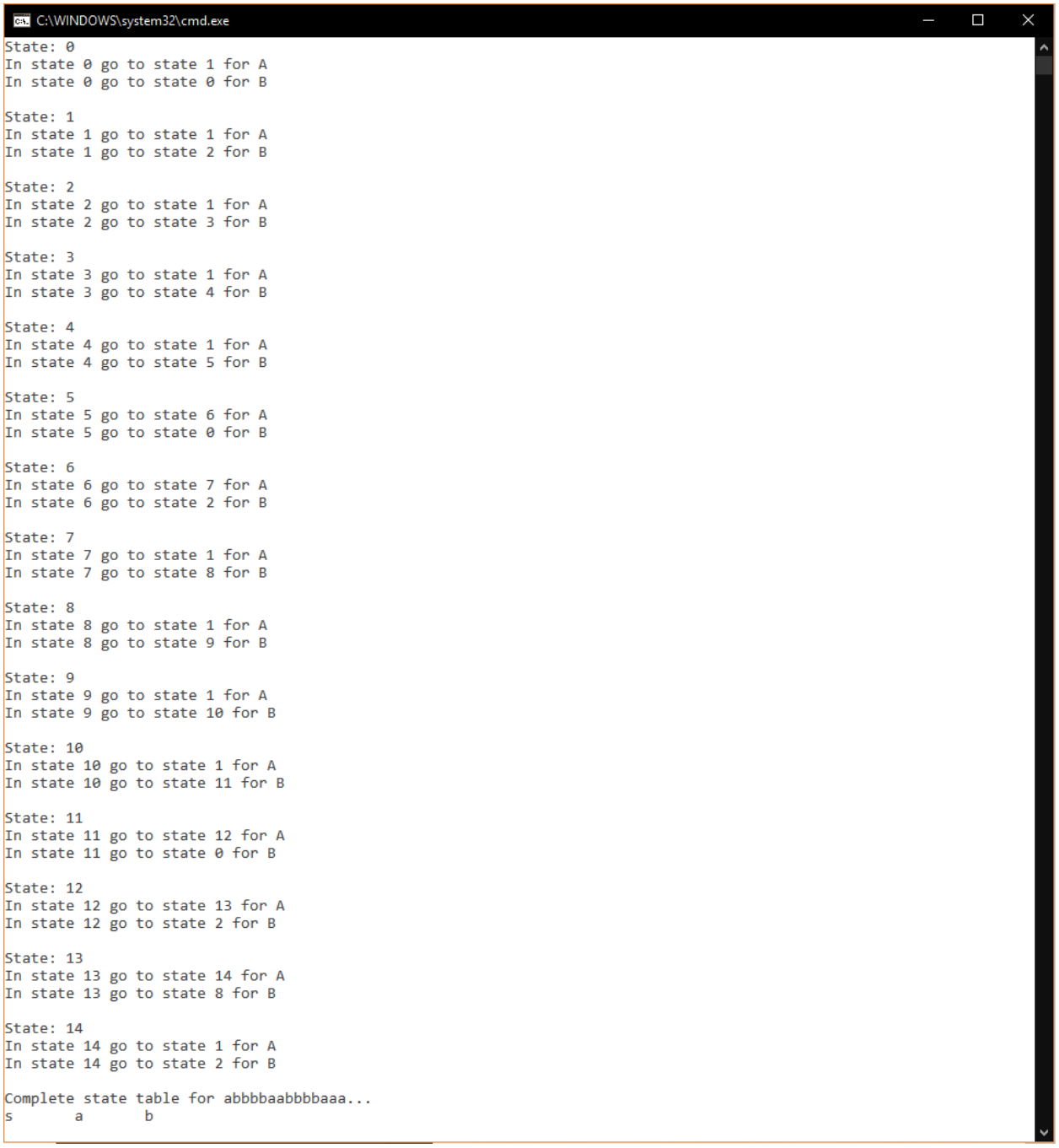


Output 1 (1 of 2)

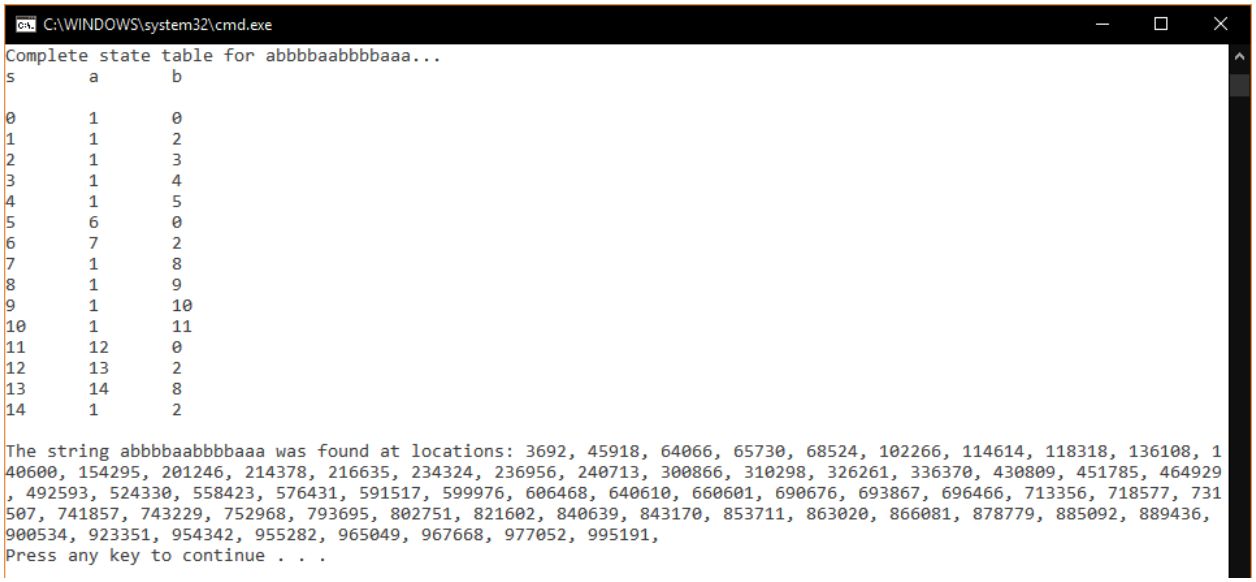


Output 1 (2 of 2)

String = “abbbbaabbbbaaa”

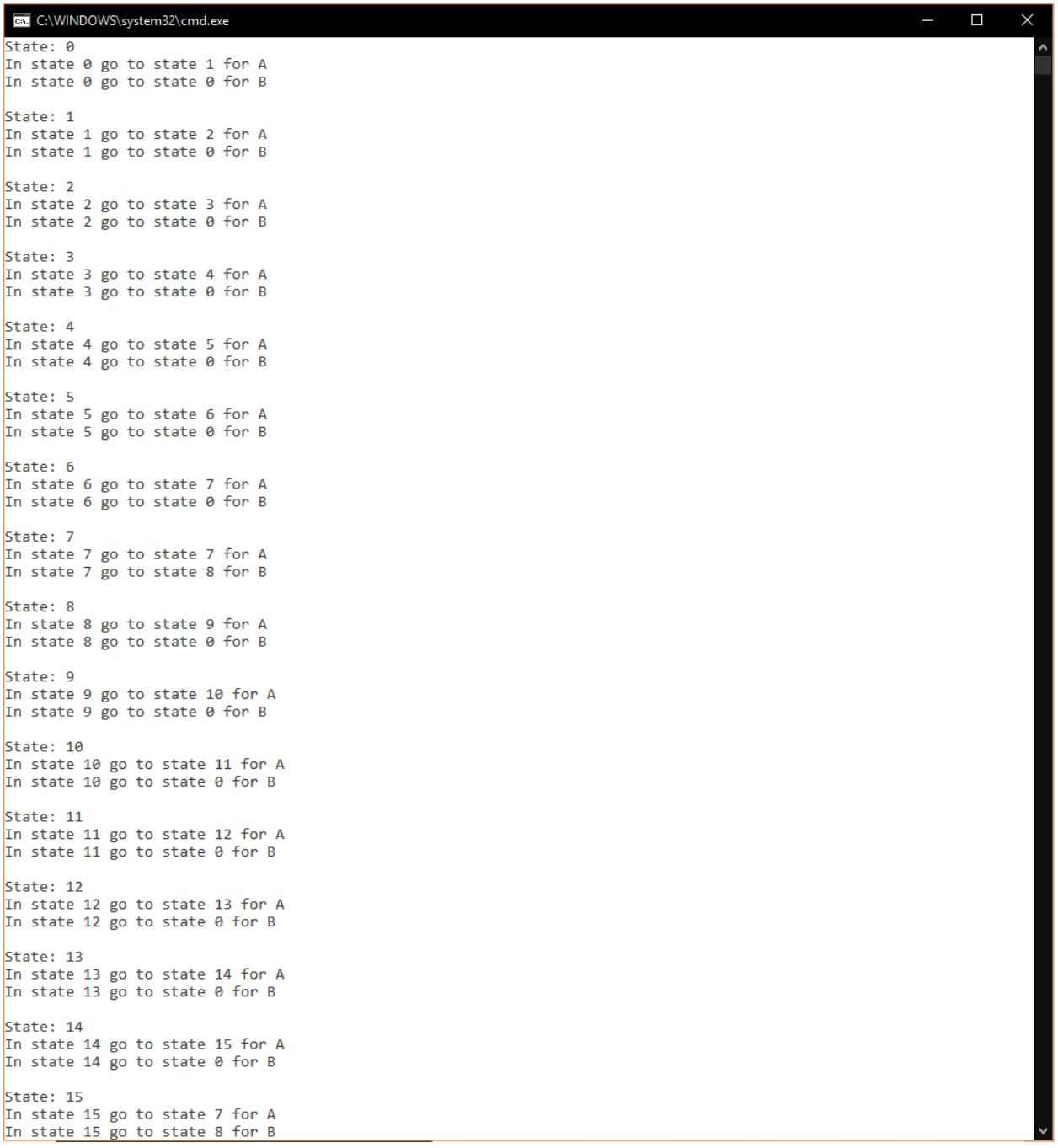


Output 2 (1 of 2)

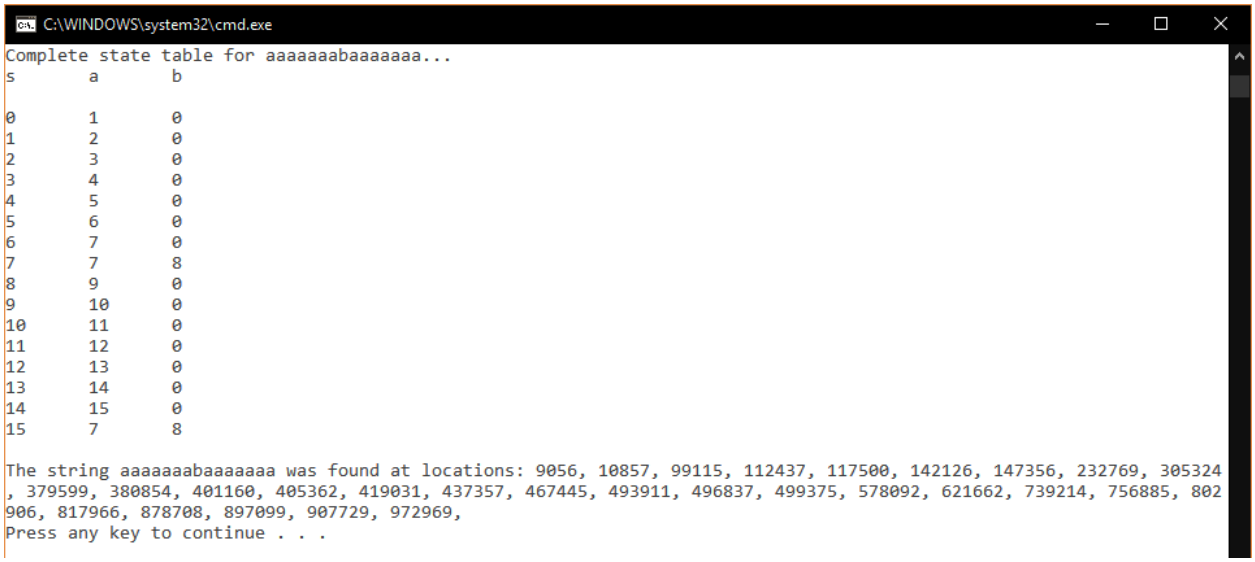


Output 2 (2 of 2)

String = “aaaaaaabaaaaaaa”

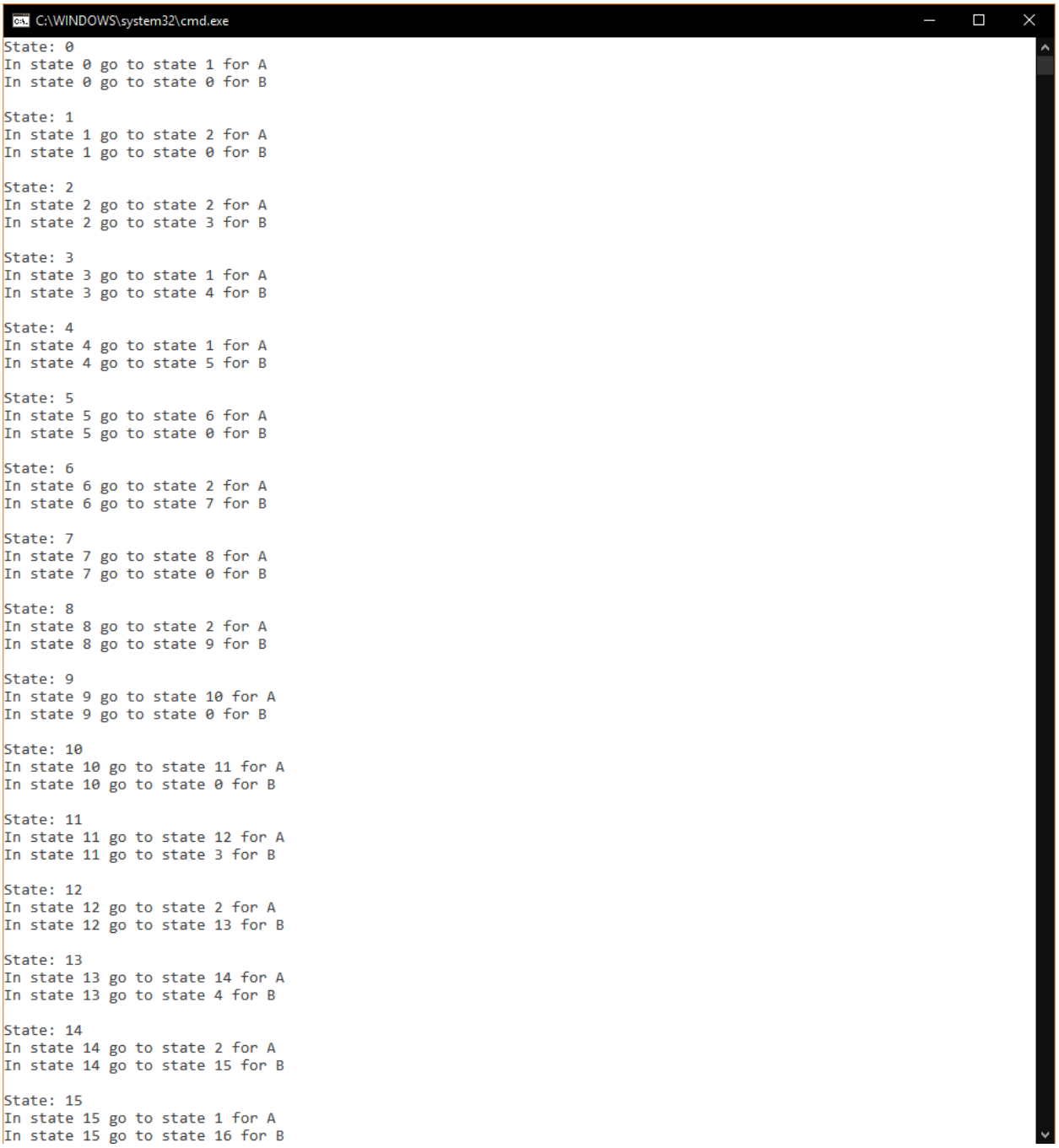


Output 3 (1 of 2)

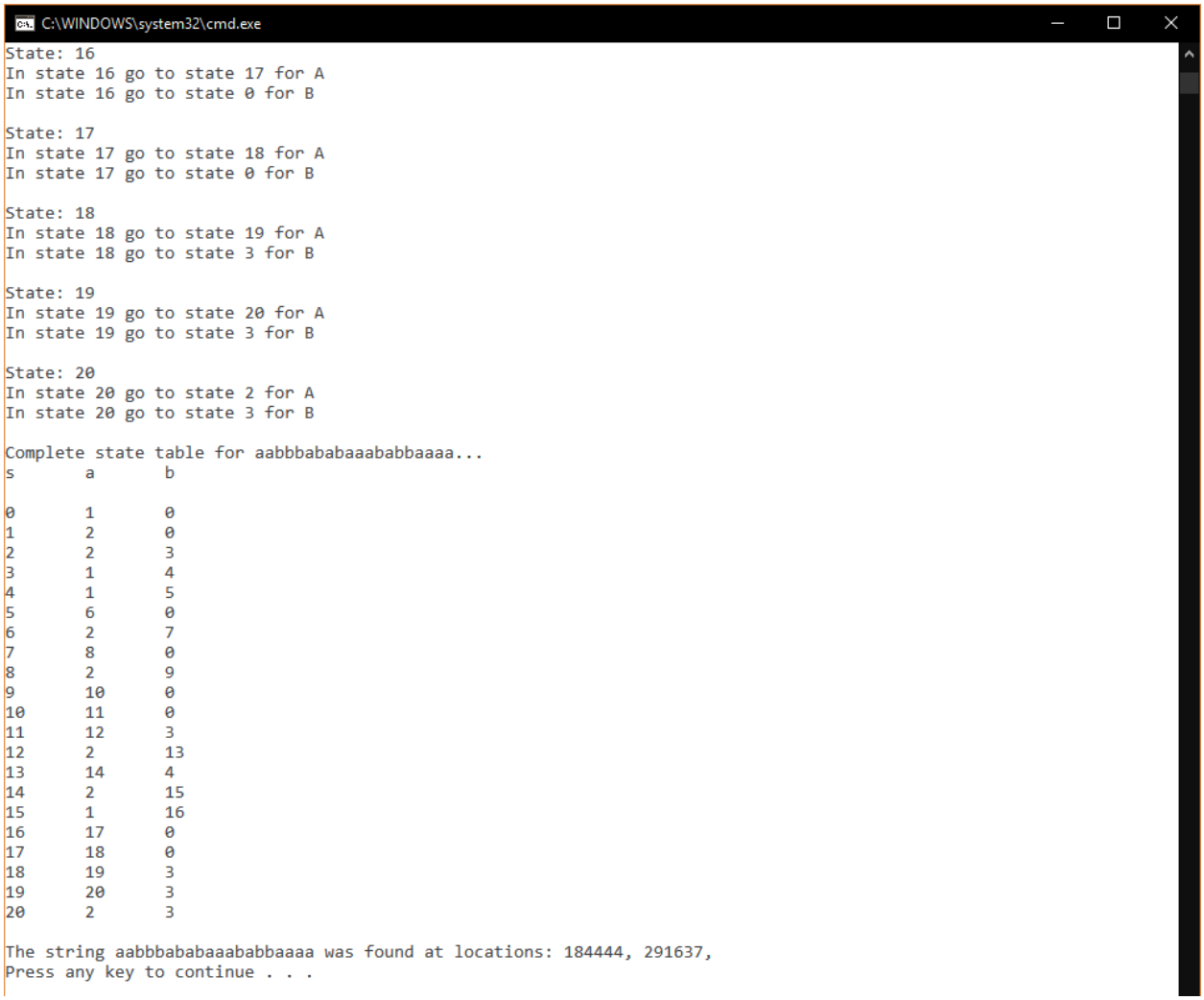


Output 3 (2 of 2)

String = “aabbbababaaababbaaaa”

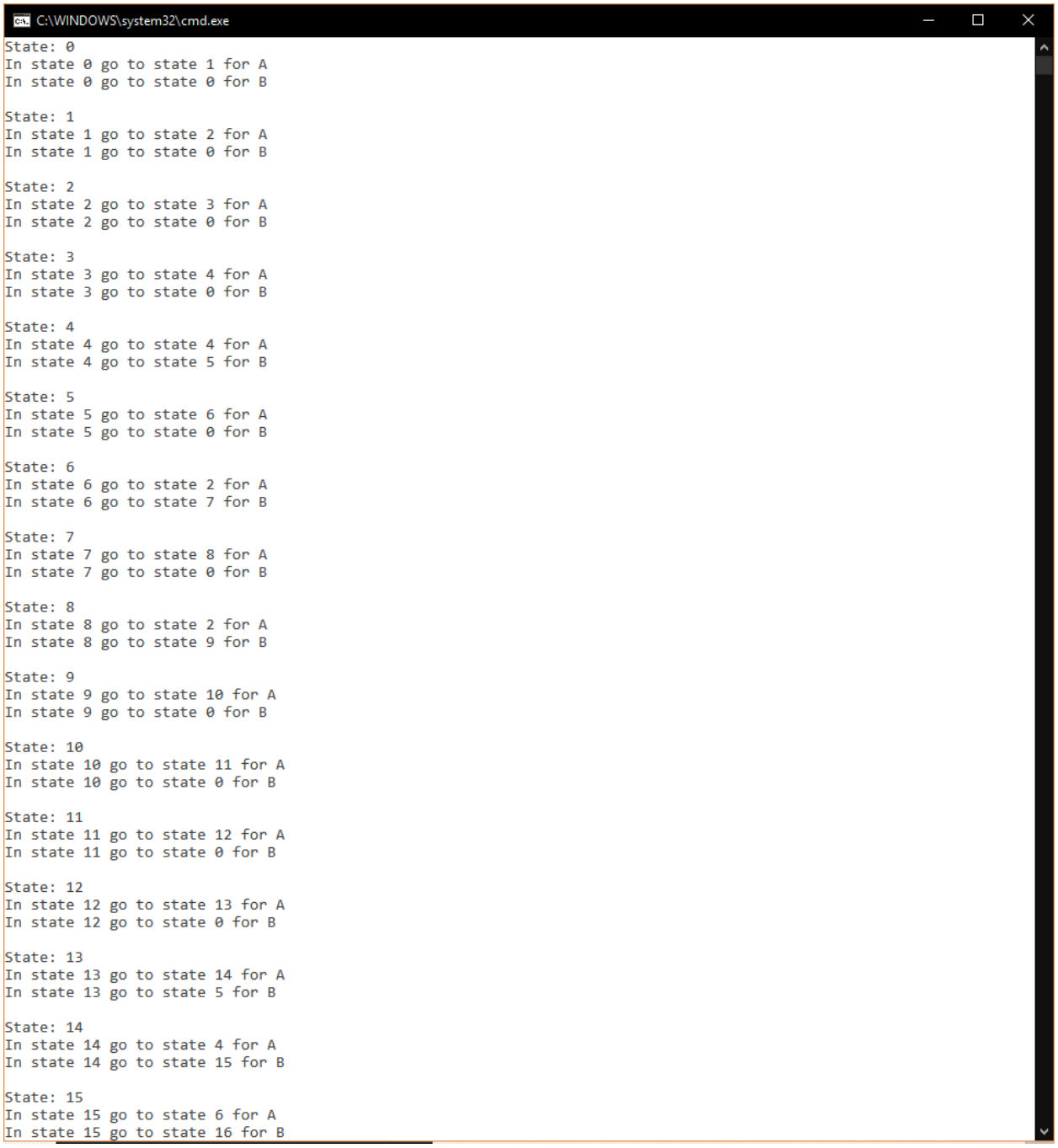


Output 4 (1 of 2)

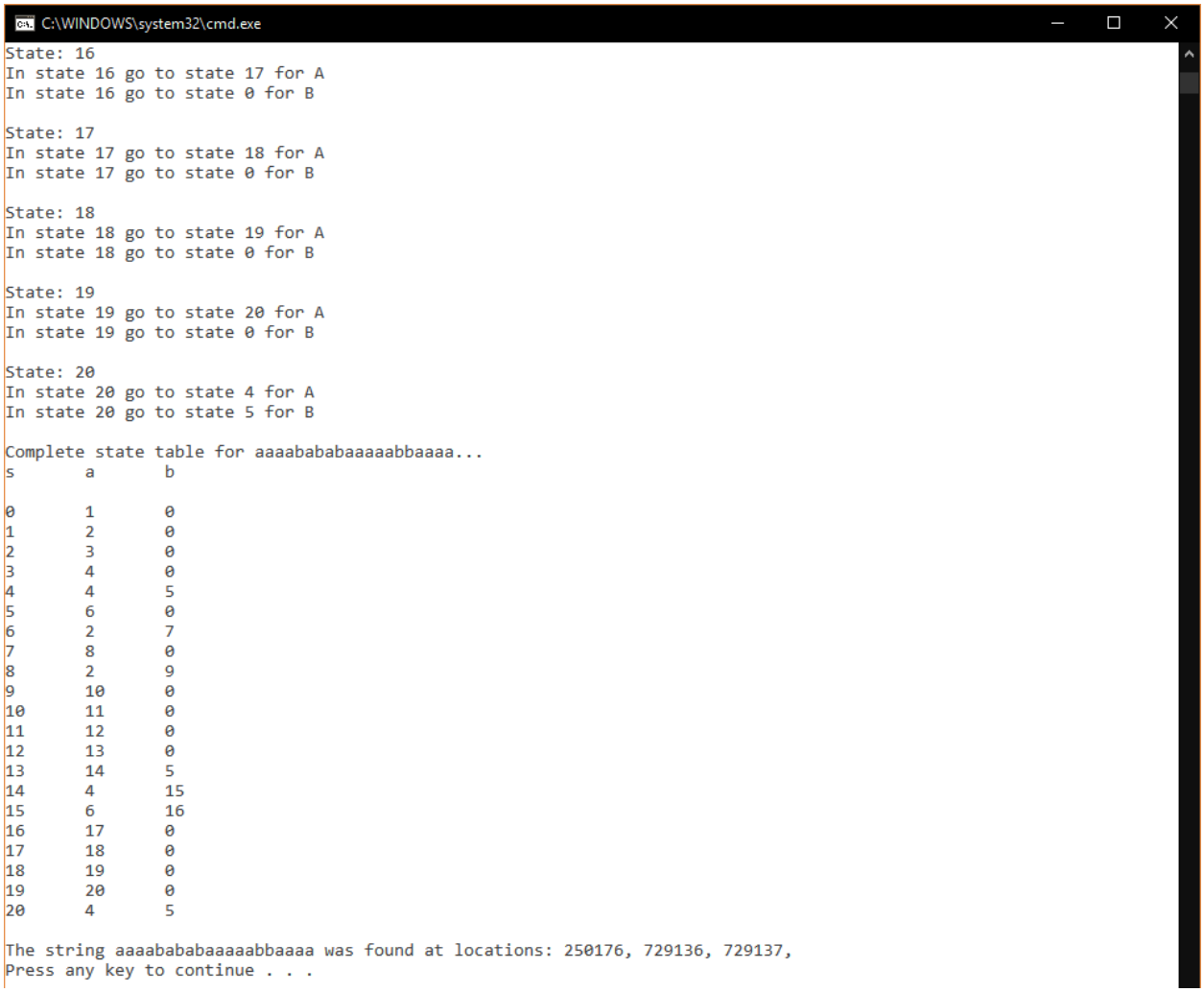


Output 4 (2 of 2)

String = “aaaababababaaaaabbaaaa”



Output 5 (1 of 2)



Output 5 (2 of 2)

Conclusion

This assignment was fun. I always enjoy working on huge files that no human would be able to process in a reasonable time. It really demonstrates the power of computers and why they are used so often. Looking forward to our next assignment!