Source File: `~/4301/09/lab09.(C|CPP|cpp|c++|cc|cxx|cp)`
Input: Under control of `main` function
Output: Under control of `main` function
Value: 2

For $\Sigma = \{a, b, =, \sqcup, 0, 1\}$, construct a Turing machine for the following:

$$\left\{ (x =, x = ans) \mid x \in \{a, b\}^*, \quad ans = \begin{cases} 1, & \text{if } x \text{ contains an odd number of occurrences of the substring } aa \\ 0, & \text{otherwise} \end{cases} \right\}$$

Note that aaa has two occurrences of aa.

A header file is shown in Figure 1, a sample `main` function for testing your implementation is shown in Figure 2, and a sample execution sequence is shown in Figure 3. To use the `Makefile` as distributed in class, add a target of `lab09` to `targets2srcfiles`.

Additional notes:

- As each input line is read in, the `main` function creates the initial tape contents as follows: ten blanks followed by the original input line followed by more blanks.
- The halt state is `0` and the start state is `1`.

```
1  #ifndef TURING_H
2  #define TURING_H
3
4  #include <iostream>
5  #include <string>
6  #include <map>
7
8  using namespace std;
9
10 class TableEntry
11 {
12  public:
13   TableEntry(char readChar, char writeChar, int move, uint nextState)
14   {
15     setReadCharacter(readChar);
16     setWriteCharacter(writeChar);
17     setMove(move);
18     setNextState(nextState);
19   }
20   void setReadCharacter(char ch)
21   {
22     readCharacter = ch;
23   }
24   void setWriteCharacter(char ch)
25   {
26     writeCharacter = ch;
27   }
```

Figure 1. `/usr/local/4301/include/turing.h` (Part 1 of 3)

```
28    void setMove(int moveAmt)
29    {
30      move = moveAmt;
31    }
32    void setNextState(uint state)
33    {
34      nextState = state;
35    }
36    char getReadCharacter() const
37    {
38      return readCharacter;
39    }
40    char getWriteCharacter() const
41    {
42      return writeCharacter;
43    }
44    int getMove() const
45    {
46      return move;
47    }
48    uint getNextState() const
49    {
50      return nextState;
51    }
52   private:
53    char readCharacter;
54    char writeCharacter;
55    int  move;
56    uint nextState;
57  };
58
59  class Turing
60  {
61   public:
62    // default constructor -- initializes private data members name,
63    // labNumber, and description
64    Turing();
65    // Member function InitializeMachine() initializes the private data
66    // member machine, a multimap where the key is the current state and
67    // the value is a class object containing the read character, write
68    // character, tape move, and next state
69    void initializeMachine();
70    // Member function OutputID() writes name, class, lab number, and
71    // lab description to output stream out
72    void outputID(ostream& out) const;
73    // Member function ImplementTuring() executes the machine on the
74    // given tape
75    void implementTuring(string& tape) const;
```

Figure 1. `/usr/local/4301/include/turing.h` (Part 2 of 3)

```
76   private:
77    string name;
78    int labNumber;
79    string description;
80    multimap<uint, TableEntry> machine;
81  };
82
83  #endif
```

Figure 1. `/usr/local/4301/include/turing.h` (Part 3 of 3)

```
1   #include <turing.h>
2
3   using namespace std;
4
5   int main()
6   {
7     Turing myTuring;
8     string dataLine, tape;
9
10    myTuring.initializeMachine();
11    myTuring.outputID(cout);
12
13    while (getline(cin, dataLine))
14    {
15      cout << "Input:  " << dataLine << endl;
16      tape = "          " + dataLine;
17      tape.resize(3 * tape.length(), ' ');
18      cout << "Output:  ";
19      myTuring.implementTuring(tape);
20      while (tape.back() == ' ')
21        tape.pop_back();
22      cout << tape << endl << endl;
23    }
24
25    return 0;
26  }
27
28  void Turing::outputID(ostream& out) const
29  {
30    out << name << endl;
31    out << "CS 4301" << endl;
32    out << "Lab " << labNumber << endl;
33    out << description << endl << endl;
34  }
35
```

Figure 2. `/usr/local/4301/src/lab09main.C` (Part 1 of 2)

```
36 void Turing::implementTuring(string& tape) const
37 {
38   int currentState = 1;
39   string::iterator tapeItr = tape.begin();
40   multimap<uint, TableEntry>::const_iterator turingItr;
41
42   while (currentState > 0)
43   {
44     // Use find to return an iterator to the first entry with a key of
45     // currentState
46     turingItr = machine.find(currentState);
47     if (turingItr != machine.end()) // found a key of currentState
48     {
49       while (turingItr != machine.upper_bound(currentState) &&
50              turingItr->second.getReadCharacter() != *tapeItr)
51         ++turingItr;
52       if (turingItr != machine.upper_bound(currentState))
53       {
54         currentState = turingItr->second.getNextState();
55         *tapeItr = turingItr->second.getWriteCharacter();
56         tapeItr += turingItr->second.getMove();
57       }
58       else
59       {
60         currentState = -1;
61         *tapeItr = '~';
62       }
63     }
64     else
65     {
66       currentState = -1;
67       *tapeItr = '~';
68     }
69   }
70 }
```

Figure 2. `/usr/local/4301/src/lab09main.C` (Part 2 of 2)

```
newuser@csunix ~> cd 4301
newuser@csunix ~/4301> ./getlab.ksh 09
  * Checking to see if a folder exists for Lab 09. . .No
  * Creating a folder for Lab 09
  * Checking to see if Lab 09 has sample input and output files. . .Yes
  * Copying input and output files for Lab 09
    from folder /usr/local/4301/data/09 to folder ./09
  * Checking to see if /usr/local/4301/src/lab09main.C exists. . .Yes
  * Copying file /usr/local/4301/src/lab09main.C to folder ./09
  * Checking to see if /usr/local/4301/include/lab09.h exists. . .No
  * Copying file /usr/local/4301/src/Makefile to folder ./09
  * Adding a target of lab09 to targets2srcfiles
  * Touching file ./09/lab09.cpp
  * Edit file ./09/lab09.cpp in Notepad++
newuser@csunix ~/4301> cd 09
newuser@csunix ~/4301/09> ls
00.dat       00.out       Makefile     lab09.cpp    lab09main.C
newuser@csunix ~/4301/09> make lab09
g++ -g -Wall -std=c++11 -c lab09main.C -I/usr/local/4301/include -I.
g++ -g -Wall -std=c++11 -c lab09.cpp -I/usr/local/4301/include -I.
g++ -o lab09 lab09main.o lab09.o -L/usr/local/4301/lib -lm
newuser@csunix ~/4301/09> cat 00.dat
aab=
aba=
baa=
babab=
aabbb=
bbbaa=
abbba=
aabbbbb=
bbbbbbbaa=
bbbabbbabbb=
=
a=
b=
aa=
ab=
ba=
bb=
aaa=
baa=
aba=
bbb=
abba=
baba=
babba=
abbbba=
bbabababa=
bbaaaabaaa=
babaaabaaa=
abaaaaaaaaaaa=
bababaababababa=
```

Figure 3. Commands to Compile, Link, & Run Lab 09 (Part 1 of 3)

```
53 newuser@csunix ~/4301/09> cat 00.dat | ./lab09
54 Your Name
55 CS 4301
56 Lab 9
57 {(x=, x=ans) | x is in {a,b}*,
58                ans = 1 if x contains an odd number of occurrences of the substring aa
59                ans = 0 otherwise}
60
61 Input:  aab=
62 Output:            aab=1
63
64 Input:  aba=
65 Output:            aba=0
66
67 Input:  baa=
68 Output:            baa=1
69
70 Input:  babab=
71 Output:            babab=0
72
73 Input:  aabbb=
74 Output:            aabbb=1
75
76 Input:  bbbaa=
77 Output:            bbbaa=1
78
79 Input:  abbba=
80 Output:            abbba=0
81
82 Input:  aabbbbb=
83 Output:            aabbbbb=1
84
85 Input:  bbbbbbbaa=
86 Output:            bbbbbbbaa=1
87
88 Input:  bbbabbbabbb=
89 Output:            bbbabbbabbb=0
90
91 Input:  =
92 Output:            =0
93
94 Input:  a=
95 Output:            a=0
96
97 Input:  b=
98 Output:            b=0
99
100 Input:  aa=
101 Output:            aa=1
102
103 Input:  ab=
104 Output:            ab=0
105
```

Figure 3. Commands to Compile, Link, & Run Lab 09 (Part 2 of 3)

```
Input:  ba=
Output:             ba=0

Input:  bb=
Output:             bb=0

Input:  aaa=
Output:             aaa=0

Input:  baa=
Output:             baa=1

Input:  aba=
Output:             aba=0

Input:  bbb=
Output:             bbb=0

Input:  abba=
Output:             abba=0

Input:  baba=
Output:             baba=0

Input:  babba=
Output:             babba=0

Input:  abbbba=
Output:             abbbba=0

Input:  bbabababa=
Output:             bbabababa=0

Input:  bbaaaabaaa=
Output:             bbaaaabaaa=1

Input:  babaaabaaa=
Output:             babaaabaaa=0

Input:  abaaaaaaaaaaa=
Output:             abaaaaaaaaaaa=0

Input:  bababaababababa=
Output:             bababaababababa=1

newuser@csunix ~/4301/09> cat 00.dat | ./lab09 > my.out
newuser@csunix ~/4301/09> diff 00.out my.out
newuser@csunix ~/4301/09>
```

Figure 3. Commands to Compile, Link, & Run Lab 09 (Part 3 of 3)