Additional 01 - Finite State Machine

Programming Assignment # 2

By
408262416
陳嬿婷

X86 Assembly
Language Fall
2020
Date Submitted: November 30,
2020

Analysis: (if asked)
Answer the questions if asked in the descriptions of the assignment.
Test Plan and Test Cases (required):
Briefly describe how you plan to test each function (if there is any) and the overall programs. For each function (if applicable) and the program you prepare some test cases. The test cases may be presented in a table form. For example,

Test Case Number	Input Values	Expected output
1	408262999	Odd count: 3 Even sum: 22 Result: 3*22 = 66
2	408250317	Odd count: 4 Even sum: 14 Result: 4*14 = 64
3	-1abc	Error (Undefined)
…		

Feedback: (required)
Informally tell us what could be improved.
Was it too difficult, or too easy?
不會很難，只是在畫FSM的時候花了一點時間思考
Was the assignment fun or challenging?
For challenging
Was there something that was unclear?
沒有Test case可以測試因為繳交範本跟小組作業為同一份
Was the project too long for the given amount of time?.
不會
Appendix A: Test Log (required)
This section should contain the **results of your testing phase**. It is not necessary that you type this section (turning in the actual log is fine, and preferred). The test log should contain the test case number, the date and time that the test was performed, the actual outputs, and the test result (either pass or fail). It is expected that not all test cases will necessarily pass on the first attempt. For any test case where the initial result is fail, there should eventually be another test that shows the passing of that test case. The test cases can be presented in table format. For example

Test Case Number	Input Values	Date &Time	Actual Output	Result
1	001-1	11/29/20 03:15 AM	The input contains even number of 0s.	Pass
2	0011-1	11/29/20 03:16 AM	The input contains even number of 0s.	Pass
3	0001-1	11/29/20 03:17 AM	The input contains odd number of 0s.	Pass
4	11001-1	11/29/20 03:18 AM	The input contains even number of 0s.	Pass

Appendix B: Source Code (required)

This section should contain a printed copy of your program, complete with all necessary documentation. ***Be sure your comments accurately describe what is going on in the code.***

```
INCLUDE Irvine32.inc
.data
    wrong BYTE "The input contains odd number of 0s.",0
    ENTER_KEY BYTE ?
    correct BYTE "The input contains even number of 0s.",0
.code
main PROC
StateA:
    call ReadInt    ;輸入
    cmp al,0
    je  StateB      ;如果是0跳到 State B
    cmp al,1
    je  StateA      ;如果是1跳到 State A
    jmp Evenn       ;如果輸入值不是0或1的話則跳到Evenn
StateB:
    call ReadInt
    cmp al,0
    je  StateA      ;如果是0跳到 State A
    cmp al,1
    je  StateB      ;如果是1跳到 State B
    jmp Oddd        ;如果輸入值不是0或1的話則跳到Oddd
;輸出有偶數個0---------------
Evenn:
    mov edx, OFFSET correct
    call writestring
    call Crlf
    exit
;輸出有奇數個0---------------
oddd:
     push  edx
     mov   edx,OFFSET wrong
     call  WriteString
     call Crlf
     pop   edx
     exit
;----------------------------
main ENDP
END main
```

```
0
0
1
1
-1
The input contains even number of 0s.
```

```
0
0
1
1
-1
The input contains even number of 0s.
```

```
1
1
0
0
1
-1
The input contains even number of 0s.
```

```
0
0
0
1
-1
The input contains odd number of 0s.
```