Source File: `lab32.asm`
Input: Standard Input
Output: Standard Output
Value: 5

Write a program that will implement a finite state machine for the following language. The alphabet of the language is $\Sigma = \{a, b\}$. Accept all strings that have an odd number of a's and an even number of b's.

You are to construct one function for this assignment. This function will implement a finite state machine for the given language. A description of the function as well as client code for testing your implementation is shown in Figure 1, and a sample execution sequence is shown in Figure 2. To use the `Makefile` as distributed in class, add a target of `lab32` to `targets2AsmFiles`.

```
[list -]
%INCLUDE "Along32.inc"
%INCLUDE "Macros_Along.inc"
[list +]

;------------------------------------------------------------------------
extern  isValid
; HLL prototype: bool isValid(char *string);
; This function receives the offset of a null-terminated array of
; characters. The function returns true if the string is valid and
; false otherwise.
; Implements the following code:
;      let n = the length of the string
;      let state = 0
;      for (i = 0; i < n; ++i)
;      {
;        print state and array[i]
;        switch (state)
;        {
;          case 0: if array[i] == ’a’
;                    state = 1
;                  else if array[i] == ’b’
;                    state = 2
;                  else
;                    state = 4
;          case 1: if array[i] == ’a’
;                    state = 0
;                  else if array[i] == ’b’
;                    state = 3
;                  else
;                    state = 4
;          case 2: if array[i] == ’a’
;                    state = 3
;                  else if array[i] == ’b’
;                    state = 0
;                  else
;                    state = 4
```

Figure 1. `/usr/local/3304/src/lab32main.asm` (Part 1 of 3)

```
;          case 3: if array[i] == 'a'
;                    state = 2
;                  else if array[i] == 'b'
;                    state = 1
;                  else
;                    state = 4
;          case 4: do nothing
;        } // end switch
;        print state
;      } // end for
;      return state == 1
; Receives: ESI = starting offset of array
; Returns:  bool in eax
;--------------------------------------------------------------------

SECTION .data
string  times   100 db 0
size    equ     $ - string
lit1    db      'Input String: ',0
lit2    db      'Status: ',0
valid   db      `Valid\n\n`,0
invalid db      `Invalid\n\n`,0

SECTION .bss
n       resd    1
h       resd    1

SECTION .text
        global  _start
_start:
        call    ReadDec         ; read an unsigned 32-bit integer
        mov     [h],eax         ; store in h

mainloop:
        cmp     dword [h], 0    ; while h > 0 do
        jle     done

        mov     edx,string      ;   read a string from standard input
        mov     ecx,size
        call    ReadString
        mov     dword [n],eax   ;   store length of input string in n

        mov     edx,lit1        ;   print the input string
        call    WriteString
        mov     edx,string
        call    WriteString
        mov     al,`\n`
        call    WriteChar

```

Figure 1. /usr/local/3304/src/lab32main.asm (Part 2 of 3)

```
        mov     esi,string      ;   call isValid function
        call    isValid

        mov     edx,lit2        ;   print the status of the string
        call    WriteString
        cmp     eax,1
        jne     printInvalid
        mov     edx,valid
        jmp     printIt
printInvalid:
        mov     edx,invalid
printIt:
        call    WriteString
        dec     dword [h]       ;   get ready for the next iteration
        jmp     mainloop        ; end while
done:
        Exit    {0}
```

Figure 1. /usr/local/3304/src/lab32main.asm (Part 3 of 3)

```
newuser@csunix ~/3304/32> cp /usr/local/3304/data/32/* .
newuser@csunix ~/3304/32> cp /usr/local/3304/src/Makefile .
newuser@csunix ~/3304/32> cp /usr/local/3304/src/lab32main.asm .
newuser@csunix ~/3304/32> touch lab29.asm
newuser@csunix ~/3304/32> make lab32
nasm -f elf32  -l lab32main.lst -o lab32main.o lab32main.asm -I/usr/local/3304/include/ -I.
nasm -f elf32  -l lab32.lst -o lab32.o lab32.asm -I/usr/local/3304/include/ -I.
ld -m elf_i386 --dynamic-linker /lib/ld-linux.so.2 -o lab32 lab32main.o lab32.o \
/usr/local/3304/src/Along32.o -lc
newuser@csunix ~/3304/32> ../irvine_test.sh lab32 01.dat
Input String: a
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          a         1
------------------------------
Status: Valid

```

Figure 2. Commands to Assemble, Link, & Run Lab 32 (Part 1 of 3)

```
Input String: b
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          b         2
------------------------------
Status: Invalid

Input String: aa
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          a         1
     1          a         0
------------------------------
Status: Invalid

Input String: ab
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          a         1
     1          b         3
------------------------------
Status: Invalid

Input String: ba
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          b         2
     2          a         3
------------------------------
Status: Invalid

Input String: bb
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          b         2
     2          b         0
------------------------------
Status: Invalid

```

Figure 2. Commands to Assemble, Link, & Run Lab 32 (Part 2 of 3)

```
Input String: aaa
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          a         1
     1          a         0
     0          a         1
------------------------------
Status: Valid

Input String: aab
------------------------------
  Current     Input     Next
   State     Symbol    State
------------------------------
     0          a         1
     1          a         0
     0          b         2
------------------------------
Status: Invalid

newuser@csunix ~/3304/32> ../irvine_test.sh lab32 01.dat > my.out
newuser@csunix ~/3304/32> diff 01.out my.out
newuser@csunix ~/3304/32> ../irvine_test.sh lab32 02.dat > my.out
newuser@csunix ~/3304/32> diff 02.out my.out
newuser@csunix ~/3304/32>
```

Figure 2. Commands to Assemble, Link, & Run Lab 32 (Part 3 of 3)