README

commands to create program:

```
nasm -f elf -F dwarf -g hw6a.asm
ld -m elf_i386 -o hw6 hw6a.o
```

Exacutable is ./hw6

Functions:

_start: Starts with the printPrompt to print out the prompt. Then reads in input with readUser.

Calls toInteger. (Changes input from string to integer)

Then it calls testBounds. (Is digit in bounds)

Then it calls toString. (Multiply input by 10 then change integer to string)

Then it calls printStr. (Prints resulting integer as a string)

finally calls exitCall. (Exits the program)

printPrompt: prints out the prompt to the command line that asks for an integer 0-99

readUser: Reads in the user input calls errMsg if the input is out of bounds by being larger then 99. If errMsg not called the call stack for _start continues by calling toInteger then toString then printStr then exitCall.

The input is stored in inStr

toInteger: checks if the value input is two digits or one. Sends to ifTrue to turn a one digit input from a string to integer. Sends to ifTwo if the input is two digits.

Assumes that the size of the input is in eax

ifTrue: Capable of turning a one digit number from a string to an int. Assumes a carriage return at the end of the string.

Stores int in esi

ifTwo: Capable of turning a two digit input from a string to an int.

Assumes a carriage return at the end.

Stores int in esi.

testBounds: Takes the stored integer in esi that it got from either ifTrue (one digit) or ifTwo(two digits) and tests if it is in the bounds of 0-99. If it is not jumps to errMsg

errMsg: prints the message if the input was out of bounds. Once this message is printed, It calls exitCall

toString: takes the input in esi and multiplies it by ten. \

Takes that integer and brings it to a string to print.

In this function edi holds an index.

Al stores the quotient after a division. Dl stores the remainder of a division.

The function jumps to firstCheck if the first digit of output is not a 0.

The function jumps to secondCheck if second division results in 0.

The final string is stored in retStr.

secondDigit: A label used to jump back into toString after incrementing the index for the first digit input of the string.

called if firstCheck is run.

edi is the index.

comeBack: label used to get back into toString after incrementing the index for possible second digit. Called if incIndex or secondCheck called

firstCheck: increments index edi if the first digit is valid in the input.

Once its incremented jumps to secondDigit to continue changing the integer to a string

SecondCheck: checks if index should be incremented. If the index is still at zero and the index should not be incremented because the first two values are zeros. If it is not incremented, it jumps to comeBack to continue changing the string to an integer.

If it should be incremented (i.e.: the second digit is valid for the output), it jumps to incIndex to increase the index.

incIndex: called to increment the index edi if the second digit is valid.

Once incremented jumps to comeBack to continue changing the integer to a string

printStr: Prints out the resulting string from toString.

The string it prints was held in retStr

exitCall: the call the exit the program.

Things that work:

1. Prints a prompt and checks if the value is 0-99. Error message printed if not.
2. Convert input string to integer
3. Multiply the integer by 10 then print to terminal.
    a. One digit
    b. Two digits
    c. Three digits
4. Exit

The images of this running and showing these tasks will be in the hw6a.pdf in this folder. When I was testing it I did not run into a case where it did not work, but I may have missed an edge case.