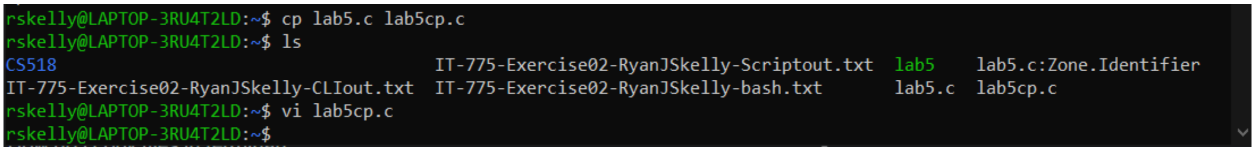
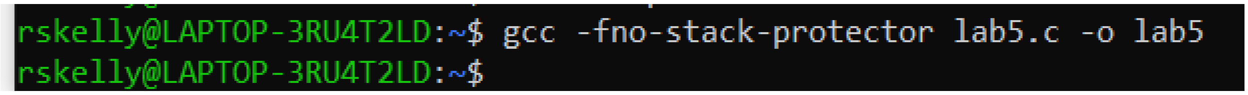
Ryan J. Skelly   Lab 5   4/13/22

1.)



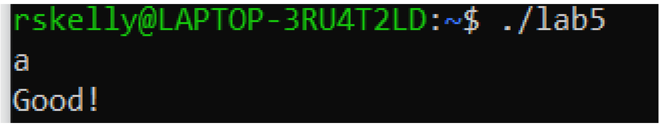
```
rskelly@LAPTOP-3RU4T2LD:~$ cp lab5.c lab5cp.c
rskelly@LAPTOP-3RU4T2LD:~$ ls
CS518                                     IT-775-Exercise02-RyanJSkelly-Scriptout.txt  lab5    lab5.c:Zone.Identifier
IT-775-Exercise02-RyanJSkelly-CLIout.txt  IT-775-Exercise02-RyanJSkelly-bash.txt       lab5.c  lab5cp.c
rskelly@LAPTOP-3RU4T2LD:~$ vi lab5cp.c
rskelly@LAPTOP-3RU4T2LD:~$
```

2.)



```
rskelly@LAPTOP-3RU4T2LD:~$ gcc -fno-stack-protector lab5.c -o lab5
rskelly@LAPTOP-3RU4T2LD:~$
```

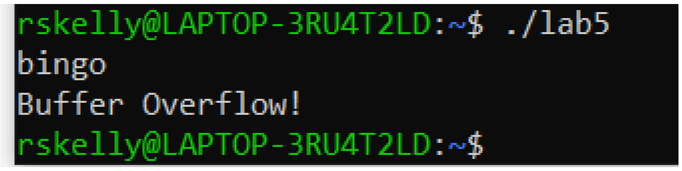
3.)



```
rskelly@LAPTOP-3RU4T2LD:~$ ./lab5
a
Good!
```

The program output “Good!” When a single lowercase a was input.

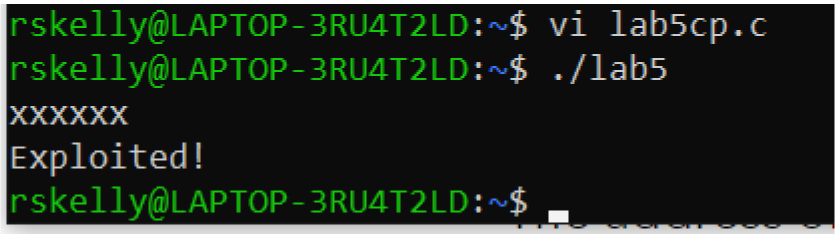
4.)



```
rskelly@LAPTOP-3RU4T2LD:~$ ./lab5
bingo
Buffer Overflow!
rskelly@LAPTOP-3RU4T2LD:~$
```

How I achieved this output was by putting ‘any’ string as the input(besides a and x)

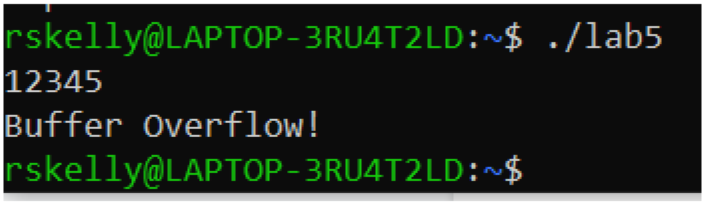
5.)



```
rskelly@LAPTOP-3RU4T2LD:~$ vi lab5cp.c
rskelly@LAPTOP-3RU4T2LD:~$ ./lab5
xxxxxx
Exploited!
rskelly@LAPTOP-3RU4T2LD:~$
```

How I achieved this output was by putting 6 lowercase xs as the input. This didnt pass as good but got caught by the null character being set to x, leading the entire memory block that the array sat into be considered as x

6.)



```
rskelly@LAPTOP-3RU4T2LD:~$ ./lab5
12345
Buffer Overflow!
rskelly@LAPTOP-3RU4T2LD:~$
```

When inputting any amount of characters greater than 4, the buffer overflows due to it not having enough space in memory to support all of the characters.