1 What if the stack grows with increasing memory addresses?

A common observation when stack overflow exploits are discussed is that the ability of overwriting the caller's return address depends on the fact that the stack grows in order of decreasing memory addresses. So, can we fix this by just changing the implementation of the system so that the stack grows upward?

Take a look at a canonical example of unsafe code below:

```
#include <string.h>

void foo (char *bar)
{
   char  buf[8];
   strcpy(buf, bar);  /* no bounds checking */
}

int main (int argc, char **argv)
{
   foo(argv[1]);
}
```

The stack during the execution of `strcpy` would look something like:

return address of foo() /* back to main() */
↑ buf[4,5,...] /* if the stack grows w/ decreasing addr */
buf[3]
↓ buf[4,5,...] /* if the stack grows w/ increasing addr */
return address of strcpy() /* back to foo() */
/* rest of strcpy's frame */ ...

What happens when we begin writing past the end of `buf` and the stack grows with increasing addresses? Is the code still exploitable?

2 Lab 3: Smoke

```
gdb bufbomb
break 136
    (or somewhere around the call to Gets())
run -u <netID>
next
    (until you enter input)
enter a bunch of 'f's = 0x66 in ASCII or some recognizable character
x /9wx buf
    (this will print out the entire buffer, find your input)
print &buf and print $rsp (notice that buf is at the top of the stack)
info frame
    (find the saved rip and where it is located)
x /40wx $rsp
    (this will print out the stack, find the saved rip)
calculate how many bytes of padding are necessary (14 blocks * 4 hex codes per block = 56 hex codes of padding)
print smoke
    (this will give you the address you need to overwrite with)
edit smoke.txt with padding, and then your return address... in LITTLE ENDIAN!!!
    (a tip for repeating characters in vim: [len]i[sequence]C-[ )
    (in emacs...: C-x ([seq]C-u[len]C-x) )
./sendstring < smoke.txt > smoke.bytes
check out what the smoke.bytes file looks like in a text editor
    (this will not entirely readable)
gdb ./bufbomb
break on the line you broke before
run -u <netID> < smoke.bytes
next
x /40wx $rsp (notice the return address has changed from before)
let it continue running... smokin'!

Note: you pass in smoke.bytes to ./bufbomb but you should submit smoke.txt
```