```c
/****************************************************************************
 * beer1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Sings "99 Bottles of Beer on the Wall."
 *
 * Demonstrates a for loop (and an opportunity for hierarchical
 * decomposition).
 ***************************************************************************/

#include <cs50.h>
#include <stdio.h>


int main(void)
{
    // ask user for number
    printf("How many bottles will there be? ");
    int n = GetInt();

    // exit upon invalid input
    if (n < 1)
    {
        printf("Sorry, that makes no sense.\n");
        return 1;
    }

    // sing the annoying song
    printf("\n");
    for (int i = n; i > 0; i--)
    {
        printf("%d bottle(s) of beer on the wall,\n", i);
        printf("%d bottle(s) of beer,\n", i);
        printf("Take one down, pass it around,\n");
        printf("%d bottle(s) of beer on the wall.\n\n", i - 1);
    }

    // exit when song is over
    printf("Wow, that's annoying.\n");
    return 0;
}
```

```c
/****************************************************************************
 * beer2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Sings "99 Bottles of Beer on the Wall."
 *
 * Demonstrates a while loop (and an opportunity for hierarchical
 * decomposition).
 ***************************************************************************/

#include <cs50.h>
#include <stdio.h>


int main(void)
{
    // ask user for number
    printf("How many bottles will there be? ");
    int n = GetInt();

    // exit upon invalid input
    if (n < 1)
    {
        printf("Sorry, that makes no sense.\n");
        return 1;
    }

    // sing the annoying song
    printf("\n");
    while (n > 0)
    {
        printf("%d bottle(s) of beer on the wall,\n", n);
        printf("%d bottle(s) of beer,\n", n);
        printf("Take one down, pass it around,\n");
        printf("%d bottle(s) of beer on the wall.\n\n", n - 1);
        n--;
    }

    // exit when song is over
    printf("Wow, that's annoying.\n");
    return 0;
}
```

```c
/****************************************************************************
 * beer3.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Sings "99 Bottles of Beer on the Wall."
 *
 * Demonstrates a condition within a for loop.
 ***************************************************************************/

#include <cs50.h>
#include <stdio.h>


int main(void)
{
    // ask user for number
    printf("How many bottles will there be? ");
    int n = GetInt();

    // exit upon invalid input
    if (n < 1)
    {
        printf("Sorry, that makes no sense.\n");
        return 1;
    }

    // sing the annoying song
    printf("\n");
    for (int i = n; i > 0; i--)
    {
        string s1 = (i == 1) ? "bottle" : "bottles";
        string s2 = (i == 2) ? "bottle" : "bottles";

        // sing verses
        printf("%d %s of beer on the wall,\n", i, s1);
        printf("%d %s of beer,\n", i, s1);
        printf("Take one down, pass it around,\n");
        printf("%d %s of beer on the wall.\n\n", i - 1, s2);
    }

    // exit when song is over
    printf("Wow, that's annoying.\n");
    return 0;
}
```

```c
/****************************************************************************
 * beer4.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Sings "99 Bottles of Beer on the Wall."
 *
 * Demonstrates hierarchical decomposition and parameter passing.
 ***************************************************************************/

#include <cs50.h>
#include <stdio.h>


// function prototype
void chorus(int b);


int main(void)
{
    // ask user for number
    printf("How many bottles will there be? ");
    int n = GetInt();

    // exit upon invalid input
    if (n < 1)
    {
        printf("Sorry, that makes no sense.\n");
        return 1;
    }

    // sing the annoying song
    printf("\n");
    for (int i = n; i > n; i--)
    {
       chorus(i);
    }

    // exit when song is over
    printf("Wow, that's annoying.\n");
    return 0;
}


/**
 * Sings about specified number of bottles.
 */
```

```
void chorus(int b)
{
    // use proper grammar
    string s1 = (b == 1) ? "bottle" : "bottles";
    string s2 = (b == 2) ? "bottle" : "bottles";

    // sing verses
    printf("%d %s of beer on the wall,\n", b, s1);
    printf("%d %s of beer,\n", b, s1);
    printf("Take one down, pass it around,\n");
    printf("%d %s of beer on the wall.\n\n", b - 1, s2);
}
```

lectures/2/src2m/buggy1.c

```c
/****************************************************************************
 * buggy1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Should print 10 asterisks but doesn't!
 * Can you find the bug?
 ***************************************************************************/

#include <stdio.h>

int main(void)
{
    for (int i = 0; i <= 10; i++)
        printf("*");
    return 0;
}
```

```c
/****************************************************************************
 * buggy2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Should print 10 asterisks, one per line, but doesn't!
 * Can you find the bug?
 ***************************************************************************/

#include <stdio.h>

int main(void)
{
    for (int i = 0; i <= 10; i++)
        printf("*");
        printf("\n");
    return 0;
}
```

lectures/2/src2m/increment.c

```c
#include <stdio.h>

// prototype
int cube(int input);

int main(void)
{
    int x = 2;
    printf("x is %d\n", x);
    x = cube(x);
    printf("x is %d\n", x);
    return 0;
}

int cube(int input)
{
   int output = input * input * input;
   return output;
}
```

lectures/2/src2m/return.c

```c
/*****************************************************************************
 * return.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Cubes a variable.
 *
 * Demonstrates use of parameter and return value.
 ****************************************************************************/

#include <stdio.h>


// function prototype
int cube(int a);


int main(void)
{
    int x = 2;
    printf("x is now %d\n", x);
    printf("Cubing...\n");
    x = cube(x);
    printf("Cubed!\n");
    printf("x is now %d\n", x);
    return 0;
}


/**
 * Cubes argument.
 */
int cube(int a)
{
    return a * a * a;
}
```