

```
1.  /*****
2.   * beer1.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Sings "99 Bottles of Beer on the Wall."
8.   *
9.   * Demonstrates a for loop (and an opportunity for hierarchical
10.  * decomposition).
11.  *****/
12.
13.  #include <cs50.h>
14.  #include <stdio.h>
15.
16.
17.  int main(void)
18.  {
19.      // ask user for number
20.      printf("How many bottles will there be? ");
21.      int n = GetInt();
22.
23.      // exit upon invalid input
24.      if (n < 1)
25.      {
26.          printf("Sorry, that makes no sense.\n");
27.          return 1;
28.      }
29.
30.      // sing the annoying song
31.      printf("\n");
32.      for (int i = n; i > 0; i--)
33.      {
34.          printf("%d bottle(s) of beer on the wall,\n", i);
35.          printf("%d bottle(s) of beer,\n", i);
36.          printf("Take one down, pass it around,\n");
37.          printf("%d bottle(s) of beer on the wall.\n\n", i - 1);
38.      }
39.
40.      // exit when song is over
41.      printf("Wow, that's annoying.\n");
42.      return 0;
43.  }
```

```
1.  /*****
2.   * beer2.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Sings "99 Bottles of Beer on the Wall."
8.   *
9.   * Demonstrates a while loop (and an opportunity for hierarchical
10.  * decomposition).
11.  *****/
12.
13. #include <cs50.h>
14. #include <stdio.h>
15.
16.
17. int main(void)
18. {
19.     // ask user for number
20.     printf("How many bottles will there be? ");
21.     int n = GetInt();
22.
23.     // exit upon invalid input
24.     if (n < 1)
25.     {
26.         printf("Sorry, that makes no sense.\n");
27.         return 1;
28.     }
29.
30.     // sing the annoying song
31.     printf("\n");
32.     while (n > 0)
33.     {
34.         printf("%d bottle(s) of beer on the wall,\n", n);
35.         printf("%d bottle(s) of beer,\n", n);
36.         printf("Take one down, pass it around,\n");
37.         printf("%d bottle(s) of beer on the wall.\n\n", n - 1);
38.         n--;
39.     }
40.
41.     // exit when song is over
42.     printf("Wow, that's annoying.\n");
43.     return 0;
44. }
```

```
1.  /*****
2.   * beer3.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Sings "99 Bottles of Beer on the Wall."
8.   *
9.   * Demonstrates a condition within a for loop.
10.  *****/
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15.
16. int main(void)
17. {
18.     // ask user for number
19.     printf("How many bottles will there be? ");
20.     int n = GetInt();
21.
22.     // exit upon invalid input
23.     if (n < 1)
24.     {
25.         printf("Sorry, that makes no sense.\n");
26.         return 1;
27.     }
28.
29.     // sing the annoying song
30.     printf("\n");
31.     for (int i = n; i > 0; i--)
32.     {
33.         string s1 = (i == 1) ? "bottle" : "bottles";
34.         string s2 = (i == 2) ? "bottle" : "bottles";
35.
36.         // sing verses
37.         printf("%d %s of beer on the wall,\n", i, s1);
38.         printf("%d %s of beer,\n", i, s1);
39.         printf("Take one down, pass it around,\n");
40.         printf("%d %s of beer on the wall.\n\n", i - 1, s2);
41.     }
42.
43.     // exit when song is over
44.     printf("Wow, that's annoying.\n");
45.     return 0;
46. }
```

```
1.  /*****
2.   * beer4.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Sings "99 Bottles of Beer on the Wall."
8.   *
9.   * Demonstrates hierarchical decomposition and parameter passing.
10.  *****/
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15.
16. // function prototype
17. void chorus(int b);
18.
19.
20. int main(void)
21. {
22.     // ask user for number
23.     printf("How many bottles will there be? ");
24.     int n = GetInt();
25.
26.     // exit upon invalid input
27.     if (n < 1)
28.     {
29.         printf("Sorry, that makes no sense.\n");
30.         return 1;
31.     }
32.
33.     // sing the annoying song
34.     printf("\n");
35.     for (int i = n; i > 0; i--)
36.     {
37.         chorus(i);
38.     }
39.
40.     // exit when song is over
41.     printf("Wow, that's annoying.\n");
42.     return 0;
43. }
44.
45.
46. /**
47.  * Sings about specified number of bottles.
48.  */
```

```
49. void chorus(int b)
50. {
51.     // use proper grammar
52.     string s1 = (b == 1) ? "bottle" : "bottles";
53.     string s2 = (b == 2) ? "bottle" : "bottles";
54.
55.     // sing verses
56.     printf("%d %s of beer on the wall,\n", b, s1);
57.     printf("%d %s of beer,\n", b, s1);
58.     printf("Take one down, pass it around,\n");
59.     printf("%d %s of beer on the wall.\n\n", b - 1, s2);
60. }
```

```
1.  /*****
2.   * buggy1.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Should print 10 asterisks but doesn't!
8.   * Can you find the bug?
9.   *****/
10.
11. #include <stdio.h>
12.
13. int main(void)
14. {
15.     for (int i = 0; i <= 10; i++)
16.         printf("*");
17.     return 0;
18. }
```

```
1.  /*****
2.   * buggy2.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Should print 10 asterisks, one per line, but doesn't!
8.   * Can you find the bug?
9.   *****/
10.
11. #include <stdio.h>
12.
13. int main(void)
14. {
15.     for (int i = 0; i <= 10; i++)
16.         printf("*");
17.         printf("\n");
18.     return 0;
19. }
```

```
1. #include <stdio.h>
2.
3. // prototype
4. int cube(int input);
5.
6. int main(void)
7. {
8.     int x = 2;
9.     printf("x is %d\n", x);
10.    x = cube(x);
11.    printf("x is %d\n", x);
12.    return 0;
13. }
14.
15. int cube(int input)
16. {
17.     int output = input * input * input;
18.     return output;
19. }
```



```
1.  /*****
2.   * return.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Cubes a variable.
8.   *
9.   * Demonstrates use of parameter and return value.
10.  *****/
11.
12.  #include <stdio.h>
13.
14.
15.  // function prototype
16.  int cube(int a);
17.
18.
19.  int main(void)
20.  {
21.      int x = 2;
22.      printf("x is now %d\n", x);
23.      printf("Cubing...\n");
24.      x = cube(x);
25.      printf("Cubed!\n");
26.      printf("x is now %d\n", x);
27.      return 0;
28.  }
29.
30.
31.  /**
32.   * Cubes argument.
33.   */
34.  int cube(int a)
35.  {
36.      return a * a * a;
37.  }
```