```
 1: /**************************************************************************
 2:  * array1.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 6:  *
 7:  * Computes a student's average across 2 quizzes.
 8:  *
 9:  * Demonstrates use of an array, a constant, and rounding.
10:  **************************************************************************/
11:
12: #include <cs50.h>
13: #include <stdio.h>
14:
15:
16: // number of quizzes per term
17: #define QUIZZES 2
18:
19:
20: int
21: main(void)
22: {
23:     float grades[QUIZZES], sum;
24:     int average, i;
25:
26:     // ask user for grades
27:     printf("\nWhat were your quiz scores?\n\n");
28:     for (i = 0; i < QUIZZES; i++)
29:     {
30:         printf("Quiz #%d of %d: ", i+1, QUIZZES);
31:         grades[i] = GetFloat();
32:     }
33:
34:     // compute average
35:     sum = 0;
36:     for (i = 0; i < QUIZZES; i++)
37:         sum += grades[i];
38:     average = (int) (sum / QUIZZES + 0.5);
39:
40:     // report average
41:     printf("\nYour average is: %d\n\n", average);
42: }
```



```
 1: /**************************************************************************
 2:  * array2.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 6:  *
 7:  * Computes a student's average across 2 quizzes.
 8:  *
 9:  * Demonstrates C's math library.
10:  **************************************************************************/
11:
12: #include <cs50.h>
13: #include <math.h>
14: #include <stdio.h>
15:
16:
17: // number of quizzes per term
18: #define QUIZZES 2
19:
20:
21: int
22: main(void)
23: {
24:     float grades[QUIZZES], sum;
25:     int average, i;
26:
27:     // ask user for grades
28:     printf("\nWhat were your quiz scores?\n\n");
29:     for (i = 0; i < QUIZZES; i++)
30:     {
31:         printf("Quiz #%d of %d: ", i+1, QUIZZES);
32:         grades[i] = GetFloat();
33:     }
34:
35:     // compute average
36:     sum = 0;
37:     for (i = 0; i < QUIZZES; i++)
38:         sum += grades[i];
39:     average = (int) round(sum / QUIZZES);
40:
41:     // report average
42:     printf("\nYour average is: %d\n\n", average);
43: }
```

beer2.c

lectures/3/src/


```
 1: /**************************************************************************
 2:  * beer2.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 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
18: main(void)
19: {
20:     // ask user for number
21:     printf("How many bottles will there be? ");
22:     int n = GetInt();
23:
24:     // exit upon invalid input
25:     if (n < 1)
26:     {
27:         printf("Sorry, that makes no sense.\n");
28:         return 1;
29:     }
30:
31:     // sing the annoying song
32:     printf("\n");
33:     while (n > 0)
34:     {
35:         printf("%d bottle(s) of beer on the wall,\n", n);
36:         printf("%d bottle(s) of beer,\n", n);
37:         printf("Take one down, pass it around,\n");
38:         printf("%d bottle(s) of beer on the wall.\n\n", n - 1);
39:         n--;
40:     }
41:
42:     // exit when song is over
43:     printf("Wow, that's annoying.\n");
44:     return 0;
45: }
```

beer3.c

lectures/3/src/


```
 1: /**************************************************************************
 2:  * beer3.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 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
17: 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:         // use proper grammar
35:         string s1 = (i == 1) ? "bottle" : "bottles";
36:         string s2 = (i == 2) ? "bottle" : "bottles";
37:
38:         // sing verses
39:         printf("%d %s of beer on the wall,\n", i, s1);
40:         printf("%d %s of beer,\n", i, s1);
41:         printf("Take one down, pass it around,\n");
42:         printf("%d %s of beer on the wall.\n\n", i - 1, s2);
43:     }
44:
45:     // exit when song is over
46:     printf("Wow, that's annoying.\n");
47:     return 0;
48: }
```

beer4.c 1/1

lectures/3/src/

```
 1: /***************************************************************************
 2:  * beer4.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 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
21: main(void)
22: {
23:     // ask user for number
24:     printf("How many bottles will there be? ");
25:     int n = GetInt();
26:
27:     // exit upon invalid input
28:     if (n < 1)
29:     {
30:         printf("Sorry, that makes no sense.\n");
31:         return 1;
32:     }
33:
34:     // sing the annoying song
35:     printf("\n");
36:     while (n)
37:         chorus(n--);
38:
39:     // exit when song is over
40:     printf("Wow, that's annoying.\n");
41:     return 0;
42: }
43:
44:
45: /*
46:  * Sings about specified number of bottles.
47:  */
48:
49: void
50: chorus(int b)
51: {
52:     // use proper grammar
53:     string s1 = (b == 1) ? "bottle" : "bottles";
54:     string s2 = (b == 2) ? "bottle" : "bottles";
55:
56:     // sing verses
57:     printf("%d %s of beer on the wall,\n", b, s1);
58:     printf("%d %s of beer,\n", b, s1);
59:     printf("Take one down, pass it around,\n");
60:     printf("%d %s of beer on the wall.\n\n", b - 1, s2);
61: }
```

sigma1.c 1/1

lectures/3/src/

```
 1: /***************************************************************************
 2:  * sigma1.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 6:  *
 7:  * Adds the numbers 1 through n.
 8:  *
 9:  * Demonstrates iteration.
10:  ***************************************************************************/
11:
12: #include <cs50.h>
13: #include <stdio.h>
14:
15:
16: // prototype
17: int sigma(int);
18:
19:
20: int
21: main(void)
22: {
23:     // ask user for a positive int
24:     int n;
25:     do
26:     {
27:         printf("Positive integer please: ");
28:         n = GetInt();
29:     }
30:     while (n < 1);
31:
32:     // compute sum of 1 through n
33:     int answer = sigma(n);
34:
35:     // report answer
36:     printf("%d\n", answer);
37: }
38:
39:
40: /*
41:  * Returns sum of 1 through m; returns 0 if m is not positive.
42:  */
43:
44: int
45: sigma(int m)
46: {
47:     // avoid risk of infinite loop
48:     if (m < 1)
49:         return 0;
50:
51:     // return sum of 1 through m
52:     int sum = 0;
53:     for (int i = 1; i <= m; i++)
54:         sum += i;
55:     return sum;
56: }
57:
```

```
 1: /**************************************************************************
 2:  * sigma2.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 6:  *
 7:  * Adds the numbers 1 through n.
 8:  *
 9:  * Demonstrates recursion.
10:  **************************************************************************/
11:
12: #include <cs50.h>
13: #include <stdio.h>
14:
15:
16: // prototype
17: int sigma(int);
18:
19:
20: int
21: main(void)
22: {
23:     // ask user for a positive int
24:     int n;
25:     do
26:     {
27:         printf("Positive integer please: ");
28:         n = GetInt();
29:     }
30:     while (n < 1);
31:
32:     // compute sum of 1 through n
33:     int answer = sigma(n);
34:
35:     // report answer
36:     printf("%d\n", answer);
37: }
38:
39:
40: /*
41:  * Returns sum of 1 through m; returns 0 if m is not positive.
42:  */
43:
44: int
45: sigma(int m)
46: {
47:     // base case
48:     if (m <= 0)
49:         return 0;
50:
51:     // recursive case
52:     else
53:         return (m + sigma(m-1));
54: }
55:
```



```
 1: /**************************************************************************
 2:  * string1.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 6:  *
 7:  * Prints a given string one character per line.
 8:  *
 9:  * Demonstrates strings as arrays of chars and use of strlen.
10:  **************************************************************************/
11:
12: #include <cs50.h>
13: #include <stdio.h>
14: #include <string.h>
15:
16:
17: int
18: main(void)
19: {
20:     // get line of text
21:     string s = GetString();
22:
23:     // print string, one character per line
24:     if (s != NULL)
25:     {
26:         for (int i = 0; i < strlen(s); i++)
27:         {
28:             char c = s[i];
29:             printf("%c\n", c);
30:         }
31:     }
32: }
```

```
 1: /****************************************************************************
 2:  * string2.c
 3:  *
 4:  * Computer Science 50
 5:  * David J. Malan
 6:  *
 7:  * Prints a given string one character per line.
 8:  *
 9:  * Demonstrates strings as arrays of chars with slight optimization.
10:  ****************************************************************************/
11:
12: #include <cs50.h>
13: #include <stdio.h>
14: #include <string.h>
15:
16:
17: int
18: main(void)
19: {
20:     // get line of text
21:     string s = GetString();
22:
23:     // print string, one character per line
24:     if (s != NULL)
25:     {
26:         for (int i = 0, n = strlen(s); i < n; i++)
27:         {
28:             printf("%c\n", s[i]);
29:         }
30:     }
31: }
```