```
 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(int argc, char * argv[])
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:  * int
42:  * sigma(int m)
43:  *
44:  * Returns sum of 1 through m; returns 0 if m is not positive.
45:  */
46:
47: int
48: sigma(int m)
49: {
50:     // avoid risk of infinite loop
51:     if (m < 1)
52:         return 0;
53:
54:     // return sum of 1 through m
55:     int sum = 0;
56:     for (int i = 1; i <= m; i++)
57:         sum += i;
58:     return sum;
59: }
60:
```

```
 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(int argc, char * argv[])
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:  * int
42:  * sigma(int m)
43:  *
44:  * Returns sum of 1 through m; returns 0 if m is not positive.
45:  */
46:
47: int
48: sigma(int m)
49: {
50:     // base case
51:     if (m <= 0)
52:         return 0;
53:
54:     // recursive case
55:     else
56:         return (m + sigma(m-1));
57: }
58:
```