

```
1. /**
2.  * noswap.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Should swap two variables' values, but doesn't! How come?
8.  */
9.
10. #include <stdio.h>
11.
12. void swap(int a, int b);
13.
14. int main(void)
15. {
16.     int x = 1;
17.     int y = 2;
18.
19.     printf("x is %i\n", x);
20.     printf("y is %i\n", y);
21.     printf("Swapping...\n");
22.     swap(x, y);
23.     printf("Swapped!\n");
24.     printf("x is %i\n", x);
25.     printf("y is %i\n", y);
26. }
27.
28. /**
29.  * Fails to swap arguments' values.
30.  */
31. void swap(int a, int b)
32. {
33.     int tmp = a;
34.     a = b;
35.     b = tmp;
36. }
```

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

```
49.     for (int i = 1; i <= m; i++)
50.     {
51.         sum += i;
52.     }
53.     return sum;
54. }
```

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

```
49.     {  
50.         return (m + sigma(m - 1));  
51.     }  
52. }
```