

```
1. /**
2.  * ascii-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Displays the mapping between alphabetical ASCII characters and
8.  * their decimal equivalents.
9.  *
10. * Demonstrates casting from int to char.
11. */
12.
13. #include <stdio.h>
14.
15. int main(void)
16. {
17.     // display mapping for uppercase letters
18.     for (int i = 65; i < 65 + 26; i++)
19.     {
20.         printf("%c: %i\n", (char) i, i);
21.     }
22.
23.     // separate uppercase from lowercase
24.     printf("\n");
25.
26.     // display mapping for lowercase letters
27.     for (int i = 97; i < 97 + 26; i++)
28.     {
29.         printf("%c: %i\n", (char) i, i);
30.     }
31. }
```

```
1. /**
2.  * ascii-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Displays the mapping between alphabetical ASCII characters and
8.  * their decimal equivalents.
9.  *
10. * Demonstrates casting from char to int.
11. */
12.
13. #include <stdio.h>
14.
15. int main(void)
16. {
17.     // display mapping for uppercase letters
18.     for (char c = 'A'; c <= 'Z'; c++)
19.     {
20.         printf("%c: %i\n", c, (int) c);
21.     }
22.
23.     // separate uppercase from lowercase
24.     printf("\n");
25.
26.     // display mapping for lowercase letters
27.     for (char c = 'a'; c <= 'z'; c++)
28.     {
29.         printf("%c: %i\n", c, (int) c);
30.     }
31. }
```

```
1. /*****
2.  * buggy-0.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. }
```

```
1.  /*****
2.   * buggy-1.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. }
```

```
1. /**
2.  * cough-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates suboptimal design (and coughing).
10. */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     // cough three times
17.     printf("cough\n");
18.     printf("cough\n");
19.     printf("cough\n");
20. }
```

```
1. /**
2.  * cough-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates better design via a loop.
10. */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     // cough three times
17.     for (int i = 0; i < 3; i++)
18.     {
19.         printf("cough\n");
20.     }
21. }
```

```
1. /**
2.  * cough-2.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates abstraction and hierarchical decomposition.
10. */
11.
12. #include <stdio.h>
13.
14. // prototype
15. void cough(void);
16.
17. int main(void)
18. {
19.     // cough three times
20.     for (int i = 0; i < 3; i++)
21.     {
22.         cough();
23.     }
24. }
25.
26. /**
27.  * Coughs once.
28.  */
29. void cough(void)
30. {
31.     printf("cough\n");
32. }
```

```
1. /**
2.  * cough-3.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates parameterization.
10. */
11.
12. #include <stdio.h>
13.
14. // prototype
15. void cough(int n);
16.
17. int main(void)
18. {
19.     // cough three times
20.     cough(3);
21. }
22.
23. /**
24.  * Coughs n times.
25.  */
26. void cough(int n)
27. {
28.     for (int i = 0; i < n; i++)
29.     {
30.         printf("cough\n");
31.     }
32. }
```



```
1. /**
2.  * cough-4.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times and sneezes three times.
8.  *
9.  * Demonstrates further abstraction.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. // prototypes
16. void cough(int n);
17. void say(string word, int n);
18. void sneeze(int n);
19.
20. int main(void)
21. {
22.     // cough three times
23.     cough(3);
24.
25.     // sneeze three times
26.     sneeze(3);
27.
28. }
29.
30. /**
31.  * Coughs n times.
32.  */
33. void cough(int n)
34. {
35.     say("cough", n);
36. }
37.
38. /**
39.  * Says word n times.
40.  */
41. void say(string word, int n)
42. {
43.     for (int i = 0; i < n; i++)
44.     {
45.         printf("%s\n", word);
46.     }
47. }
48.
```

```
49. /**
50.  * Sneezes n times.
51.  */
52. void sneeze(int n)
53. {
54.     say("achoo", n);
55. }
```

```
1. /**
2.  * function-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Prints a user's name.
8.  *
9.  * Demonstrates a function (not from a library) with a side effect.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. // prototype
16. void PrintName(string name);
17.
18. int main(void)
19. {
20.     printf("Your name: ");
21.     string s = GetString();
22.     PrintName(s);
23. }
24.
25. /**
26.  * Says hello to someone by name.
27.  */
28. void PrintName(string name)
29. {
30.     printf("hello, %s\n", name);
31. }
```

```
1. /**
2.  * function-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Demands that user provide a positive integer.
8.  *
9.  * Demonstrates use of a function (not from a library) with a return value.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. // prototype
16. int GetPositiveInt();
17.
18. int main(void)
19. {
20.     int n = GetPositiveInt();
21.     printf("Thanks for the %i!\n", n);
22. }
23.
24. /**
25.  * Gets a positive integer from a user.
26.  */
27. int GetPositiveInt(void)
28. {
29.     int n;
30.     do
31.     {
32.         printf("Please give me a positive int: ");
33.         n = GetInt();
34.     }
35.     while (n < 1);
36.     return n;
37. }
```

```
1. /**
2.  * string-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Prints a 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. int main(void)
17. {
18.     // get line of text
19.     string s = GetString();
20.
21.     // print string, one character per line
22.     for (int i = 0; i < strlen(s); i++)
23.     {
24.         printf("%c\n", s[i]);
25.     }
26. }
```

```
1. /**
2.  * string-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Prints a string, one character per line.
8.  *
9.  * Demonstrates error checking.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14. #include <string.h>
15.
16. int main(void)
17. {
18.     // get line of text
19.     string s = GetString();
20.
21.     // print string, one character per line
22.     if (s != NULL)
23.     {
24.         for (int i = 0; i < strlen(s); i++)
25.         {
26.             printf("%c\n", s[i]);
27.         }
28.     }
29. }
```

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