

```
1. /**
2.  * ages.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Ages people by a year.
8.  *
9.  * Demonstrates arrays.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(void)
16. {
17.     // determine number of people
18.     int n;
19.     do
20.     {
21.         printf("Number of people in room: ");
22.         n = GetInt();
23.     }
24.     while (n < 1);
25.
26.     // declare array in which to store everyone's age
27.     int ages[n];
28.
29.     // get everyone's age
30.     for (int i = 0; i < n; i++)
31.     {
32.         printf("Age of person #%i: ", i + 1);
33.         ages[i] = GetInt();
34.     }
35.
36.     // report everyone's age a year hence
37.     printf("Time passes...\n");
38.     for (int i = 0; i < n; i++)
39.     {
40.         printf("A year from now, person #%i will be %i years old.\n", i + 1, ages[i] + 1);
41.     }
42. }
```

```
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.  * capitalize-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Capitalizes a given string.
8.  *
9.  * Demonstrates casting and iteration over strings as arrays of chars.
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.     // capitalize text
22.     for (int i = 0, n = strlen(s); i < n; i++)
23.     {
24.         if (s[i] >= 'a' && s[i] <= 'z')
25.         {
26.             printf("%c", s[i] - ('a' - 'A'));
27.         }
28.         else
29.         {
30.             printf("%c", s[i]);
31.         }
32.     }
33.     printf("\n");
34. }
```

```
1. /**
2.  * capitalize-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Capitalizes a given string.
8.  *
9.  * Demonstrates islower and toupper.
10. */
11.
12. #include <cs50.h>
13. #include <ctype.h>
14. #include <stdio.h>
15. #include <string.h>
16.
17. int main(void)
18. {
19.     // get line of text
20.     string s = GetString();
21.
22.     // capitalize text
23.     for (int i = 0, n = strlen(s); i < n; i++)
24.     {
25.         if (islower(s[i]))
26.         {
27.             printf("%c", toupper(s[i]));
28.         }
29.         else
30.         {
31.             printf("%c", s[i]);
32.         }
33.     }
34.     printf("\n");
35. }
```

```
1. /**
2.  * capitalize-2.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Capitalizes a given string.
8.  *
9.  * Demonstrates further simplification of code with toupper.
10. */
11.
12. #include <cs50.h>
13. #include <ctype.h>
14. #include <stdio.h>
15. #include <string.h>
16.
17. int main(void)
18. {
19.     // get line of text
20.     string s = GetString();
21.
22.     // capitalize text
23.     for (int i = 0, n = strlen(s); i < n; i++)
24.     {
25.         printf("%c", toupper(s[i]));
26.     }
27.     printf("\n");
28. }
```

```
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. }
```