```c
/**
 * ages.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Ages people by a year.
 *
 * Demonstrates arrays.
 */

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    // determine number of people
    int n;
    do
    {
        printf("Number of people in room: ");
        n = GetInt();
    }
    while (n < 1);

    // declare array in which to store everyone's age
    int ages[n];

    // get everyone's age
    for (int i = 0; i < n; i++)
    {
        printf("Age of person #%i: ", i + 1);
        ages[i] = GetInt();
    }

    // report everyone's age a year hence
    printf("Time passes...\n");
    for (int i = 0; i < n; i++)
    {
        printf("A year from now, person #%i will be %i years old.\n", i + 1, ages[i] + 1);
    }
}
```

```c
/**
 * argv-0.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints program's first command-line argument; assumes it's present.
 *
 * Demonstrates use of argv.
 */

#include <cs50.h>
#include <stdio.h>

int main(int argc, string argv[])
{
    printf("%s\n", argv[1]);
}
```

```c
/**
 * argv-1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints command-line arguments, one per line.
 *
 * Demonstrates use of argv.
 */

#include <cs50.h>
#include <stdio.h>

int main(int argc, string argv[])
{
    // print arguments
    for (int i = 0; i < argc; i++)
    {
        printf("%s\n", argv[i]);
    }
}
```

```c
/**
 * argv-2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints command-line arguments, one character per line.
 *
 * Demonstrates argv as a two-dimensional array.
 */

#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(int argc, string argv[])
{
    // print arguments
    for (int i = 0; i < argc; i++)
    {
        for (int j = 0, n = strlen(argv[i]); j < n; j++)
        {
            printf("%c\n", argv[i][j]);
        }
    }
}
```

```c
/**
 * capitalize-0.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Capitalizes a given string.
 *
 * Demonstrates casting and iteration over strings as arrays of chars.
 */

#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // capitalize text
    for (int i = 0, n = strlen(s); i < n; i++)
    {
        if (s[i] >= 'a' && s[i] <= 'z')
        {
            printf("%c", s[i] - ('a' - 'A'));
        }
        else
        {
            printf("%c", s[i]);
        }
    }
    printf("\n");
}
```

```c
/**
 * capitalize-1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Capitalizes a given string.
 *
 * Demonstrates islower and toupper.
 */

#include <cs50.h>
#include <ctype.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // capitalize text
    for (int i = 0, n = strlen(s); i < n; i++)
    {
        if (islower(s[i]))
        {
            printf("%c", toupper(s[i]));
        }
        else
        {
            printf("%c", s[i]);
        }
    }
    printf("\n");
}
```

```c
/**
 * capitalize-2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Capitalizes a given string.
 *
 * Demonstrates further simplification of code with toupper.
 */

#include <cs50.h>
#include <ctype.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // capitalize text
    for (int i = 0, n = strlen(s); i < n; i++)
    {
        printf("%c", toupper(s[i]));
    }
    printf("\n");
}
```

lectures/2/w/src2w/function-0.c

```c
/**
 * function-0.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a user's name.
 *
 * Demonstrates a function (not from a library) with a side effect.
 */

#include <cs50.h>
#include <stdio.h>

// prototype
void PrintName(string name);

int main(void)
{
    printf("Your name: ");
    string s = GetString();
    PrintName(s);
}

/**
 * Says hello to someone by name.
 */
void PrintName(string name)
{
    printf("hello, %s\n", name);
}
```

```c
/**
 * function-1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Demands that user provide a positive integer.
 *
 * Demonstrates use of a function (not from a library) with a return value.
 */

#include <cs50.h>
#include <stdio.h>

// prototype
int GetPositiveInt();

int main(void)
{
    int n = GetPositiveInt();
    printf("Thanks for the positive int!\n", n);
}

/**
 * Gets a positive integer from a user.
 */
int GetPositiveInt(void)
{
    int n;
    do
    {
        printf("Please give me a positive in: ");
        n = GetInt();
    }
    while (n < 1);
    return n;
}
```

```c
/**
 * string-0.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a string, one character per line.
 *
 * Demonstrates strings as arrays of chars and use of strlen.
 */

#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // print string, one character per line
    for (int i = 0; i < strlen(s); i++)
    {
        printf("%c\n", s[i]);
    }
}
```

```c
/**
 * string-1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a string, one character per line.
 *
 * Demonstrates error checking.
 */

#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // print string, one character per line
    if (s != NULL)
    {
        for (int i = 0; i < strlen(s); i++)
        {
            printf("%c\n", s[i]);
        }
    }
}
```

```c
/**
 * string-2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a string, one character per line.
 *
 * Demonstrates optimization of a loop.
 */

#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // print string, one character per line
    if (s != NULL)
    {
        for (int i = 0, n = strlen(s); i < n; i++)
        {
            printf("%c\n", s[i]);
        }
    }
}
```

```c
/****************************************************************************
 * string.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a given string one character per line.
 *
 * Demonstrates strings as arrays of chars and use of strlen.
 ***************************************************************************/

#include <cs50.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
    // get line of text
    string s = GetString();

    // print string, one character per line
    if (s != NULL)
    {
        for (int i = 0, n = strlen(s); i < n; i++)
        {
            printf("%c\n", s[i]);
        }
    }
}
```