```c
/****************************************************************************
 * argv1.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
    printf("\n");
    for (int i = 0; i < argc; i++)
        printf("%s\n", argv[i]);
    printf("\n");
    return 0;
}
```

```c
/****************************************************************************
 * argv2.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
    printf("\n");
    for (int i = 0; i < argc; i++)
    {
        for (int j = 0, n = strlen(argv[i]); j < n; j++)
            printf("%c\n", argv[i][j]);
        printf("\n");
    }
    return 0;
}
```

```c
/*****************************************************************************
 * array.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Computes a student's average across 2 quizzes.
 *
 * Demonstrates C's math library.
 *****************************************************************************/

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

// number of quizzes per term
#define QUIZZES 2

int main(void)
{
    // ask user for grades
    float grades[QUIZZES];
    printf("\nWhat were your quiz scores?\n\n");
    for (int i = 0; i < QUIZZES; i++)
    {
        printf("Quiz #%d of %d: ", i+1, QUIZZES);
        grades[i] = GetFloat();
    }

    // compute average
    float sum = 0;
    for (int i = 0; i < QUIZZES; i++)
        sum += grades[i];
    int average = (int) round(sum / QUIZZES);

    // report average
    printf("\nYour average is: %d\n\n", average);

    return 0;
}
```

```c
/****************************************************************************
 * ascii.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Displays the mapping between alphabetical ASCII characters and
 * their decimal equivalents using one column.
 *
 * Demonstrates casting from int to char.
 ****************************************************************************/

#include <stdio.h>

int main(void)
{
    // display mapping for uppercase letters
    for (int i = 65; i < 65 + 26; i++)
        printf("%c: %d\n", (char) i, i);

    // separate uppercase from lowercase
    printf("\n");

    // display mapping for lowercase letters
    for (int i = 97; i < 97 + 26; i++)
        printf("%c: %d\n", (char) i, i);

    return 0;
}
```

```c
/****************************************************************************
 * string1.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
    for (int i = 0; i < strlen(s); i++)
    {
        char c = s[i];
        printf("%c\n", c);
    }

    return 0;
}
```

```c
/****************************************************************************
 * string2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a given string one character per line.
 *
 * Demonstrates strings as arrays of chars with slight optimization.
 ****************************************************************************/

#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, n = strlen(s); i < n; i++)
    {
        printf("%c\n", s[i]);
    }

    return 0;
}
```