```c
/****************************************************************************
 * compare2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Compares two strings.
 *
 * Demonstrates strings as pointers to characters.
 ****************************************************************************/

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


int main(void)
{
    // get line of text
    printf("Say something: ");
    char* s = GetString();

    // get another line of text
    printf("Say something: ");
    char* t = GetString();

    // try to compare strings
    if (s != NULL && t != NULL)
    {
        if (strcmp(s, t) == 0)
            printf("You typed the same thing!\n");
        else
            printf("You typed different things!\n");
    }

    return 0;
}
```

```c
/****************************************************************************
 * copy1.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Tries and fails to copy two strings.
 *
 * Demonstrates strings as pointers to arrays.
 ****************************************************************************/

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

int main(void)
{
    // get line of text
    printf("Say something: ");
    string s = GetString();
    if (s == NULL)
        return 1;

    // try (and fail) to copy string
    string t = s;

    // change "copy"
    printf("Capitalizing copy...\n");
    if (strlen(t) > 0)
        t[0] = toupper(t[0]);

    // print original and "copy"
    printf("Original: %s\n", s);
    printf("Copy:     %s\n", t);

    return 0;
}
```

```c
/****************************************************************************
 * copy2.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Copies a string.
 *
 * Demonstrates strings as pointers to arrays.
 ****************************************************************************/

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

int main(void)
{
    // get line of text
    printf("Say something: ");
    char* s = GetString();
    if (s == NULL)
        return 1;

    // allocate enough space for copy
    char* t = malloc((strlen(s) + 1) * sizeof(char));
    if (t == NULL)
        return 1;

    // copy string
    int n = strlen(s);
    for (int i = 0; i < n; i++)
        t[i] = s[i];
    t[n] = '\0';

    // change copy
    printf("Capitalizing copy...\n");
    if (strlen(t) > 0)
        t[0] = toupper(t[0]);

    // print original and copy
    printf("Original: %s\n", s);
    printf("Copy:     %s\n", t);

    // free memory
    free(s);
    free(t);
```

```
49.
50.     return 0;
51. }
```

```c
/****************************************************************************
 * pointers.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Prints a given string one character per line.
 *
 * Demonstrates pointer arithmetic.
 ****************************************************************************/

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

int main(void)
{
    // get line of text
   char* s = GetString();
    if (s == NULL)
        return 1;

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

    // free string
    free(s);

    return 0;
}
```

```c
/*****************************************************************************
 * swap.c
 *
 * David J. Malan
 * malan@harvard.edu
 *
 * Swaps two variables' values.
 *
 * Demonstrates passing by reference.
 *****************************************************************************/

#include <stdio.h>

// function prototype
void swap(int* a, int* b);

int main(void)
{
    int x = 1;
    int y = 2;

    printf("x is %d\n", x);
    printf("y is %d\n", y);
    printf("Swapping...\n");
    swap(&x, &y);
    printf("Swapped!\n");
    printf("x is %d\n", x);
    printf("y is %d\n", y);

    return 0;
}


/**
 * Swap arguments' values.
 */
void swap(int* a, int* b)
{
    int tmp = *a;
    *a = *b;
    *b = tmp;
}
```