

```
1.  /*****
2.   * compare2.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Compares two strings.
8.   *
9.   * Demonstrates strings as pointers to characters.
10.  *****/
11.
12.  #include <cs50.h>
13.  #include <stdio.h>
14.  #include <string.h>
15.
16.
17.  int main(void)
18.  {
19.      // get line of text
20.      printf("Say something: ");
21.      char* s = GetString();
22.
23.      // get another line of text
24.      printf("Say something: ");
25.      char* t = GetString();
26.
27.      // try to compare strings
28.      if (s != NULL && t != NULL)
29.      {
30.          if (strcmp(s, t) == 0)
31.              printf("You typed the same thing!\n");
32.          else
33.              printf("You typed different things!\n");
34.      }
35.
36.      return 0;
37.  }
```

```
1.  /*****
2.   * copy1.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Tries and fails to copy two strings.
8.   *
9.   * Demonstrates strings as pointers to arrays.
10.  *****/
11.
12. #include <cs50.h>
13. #include <ctype.h>
14. #include <stdio.h>
15. #include <stdlib.h>
16. #include <string.h>
17.
18. int main(void)
19. {
20.     // get line of text
21.     printf("Say something: ");
22.     string s = GetString();
23.     if (s == NULL)
24.         return 1;
25.
26.     // try (and fail) to copy string
27.     string t = s;
28.
29.     // change "copy"
30.     printf("Capitalizing copy...\n");
31.     if (strlen(t) > 0)
32.         t[0] = toupper(t[0]);
33.
34.     // print original and "copy"
35.     printf("Original: %s\n", s);
36.     printf("Copy:      %s\n", t);
37.
38.     return 0;
39. }
```

```
1.  /*****
2.   * copy2.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Copies a string.
8.   *
9.   * Demonstrates strings as pointers to arrays.
10.  *****/
11.
12. #include <cs50.h>
13. #include <ctype.h>
14. #include <stdio.h>
15. #include <stdlib.h>
16. #include <string.h>
17.
18. int main(void)
19. {
20.     // get line of text
21.     printf("Say something: ");
22.     char* s = GetString();
23.     if (s == NULL)
24.         return 1;
25.
26.     // allocate enough space for copy
27.     char* t = malloc((strlen(s) + 1) * sizeof(char));
28.     if (t == NULL)
29.         return 1;
30.
31.     // copy string
32.     int n = strlen(s);
33.     for (int i = 0; i < n; i++)
34.         t[i] = s[i];
35.     t[n] = '\0';
36.
37.     // change copy
38.     printf("Capitalizing copy...\n");
39.     if (strlen(t) > 0)
40.         t[0] = toupper(t[0]);
41.
42.     // print original and copy
43.     printf("Original: %s\n", s);
44.     printf("Copy:      %s\n", t);
45.
46.     // free memory
47.     free(s);
48.     free(t);
```

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

```
1.  /*****
2.   * pointers.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Prints a given string one character per line.
8.   *
9.   * Demonstrates pointer arithmetic.
10.  *****/
11.
12. #include <cs50.h>
13. #include <stdio.h>
14. #include <stdlib.h>
15. #include <string.h>
16.
17. int main(void)
18. {
19.     // get line of text
20.     char* s = GetString();
21.     if (s == NULL)
22.         return 1;
23.
24.     // print string, one character per line
25.     for (int i = 0, n = strlen(s); i < n; i++)
26.         printf("%c\n", *(s+i));
27.
28.     // free string
29.     free(s);
30.
31.     return 0;
32. }
```

```
1.  /*****
2.   * swap.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Swaps two variables' values.
8.   *
9.   * Demonstrates passing by reference.
10.  *****/
11.
12. #include <stdio.h>
13.
14. // function prototype
15. void swap(int* a, int* b);
16.
17. int main(void)
18. {
19.     int x = 1;
20.     int y = 2;
21.
22.     printf("x is %d\n", x);
23.     printf("y is %d\n", y);
24.     printf("Swapping...\n");
25.     swap(&x, &y);
26.     printf("Swapped!\n");
27.     printf("x is %d\n", x);
28.     printf("y is %d\n", y);
29.
30.     return 0;
31. }
32.
33.
34. /**
35.  * Swap arguments' values.
36.  */
37. void swap(int* a, int* b)
38. {
39.     int tmp = *a;
40.     *a = *b;
41.     *b = tmp;
42. }
```