```
1  // Prints an integer
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7   int n = 50;
8   printf("%i\n", n);
9 }
```

```
// Prints an integer's address

#include <stdio.h>

int main(void)

f int n = 50;
printf("%p\n", &n);
}
```

```
1  // Stores and prints a string's address via pointer arithmetic
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      char *s = "EMMA";
8      printf("%c\n", *s);
9      printf("%c\n", *(s+1));
10      printf("%c\n", *(s+2));
11      printf("%c\n", *(s+3));
12  }
```

```
// Prints an integer via its address

#include <stdio.h>

int main(void)

{
   int n = 50;
   printf("%i\n", *&n);
}
```

```
1  // Stores and prints an integer's address
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    int n = 50;
8    int *p = &n;
9    printf("%p\n", p);
10 }
```

```
1  // Stores and prints an integer via its address
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    int n = 50;
8    int *p = &n;
9    printf("%i\n", *p);
10 }
```

```
1  // Prints a string
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8     string s = "EMMA";
9     printf("%s\n", s);
10 }
```

```
1  // Prints a string's address
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8     string s = "EMMA";
9     printf("%p\n", s);
10 }
```

```
// Prints a string's address as well the addresses of its chars
 2
 3
4
      #include <cs50.h>
       #include <stdio.h>
 5
 6
       int main(void)
 8
             string s = "EMMA";
            printf("%p\n", s);
printf("%p\n", &s[0]);
printf("%p\n", &s[1]);
printf("%p\n", &s[2]);
printf("%p\n", &s[3]);
 9
10
11
12
13
14
             printf("%p\n", &s[4]);
15
      }
```

```
1  // Prints a string's chars
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8     string s = "EMMA";
9     printf("%c\n", s[0]);
10     printf("%c\n", s[1]);
11     printf("%c\n", s[2]);
12     printf("%c\n", s[3]);
13 }
```

```
1  // Stores and prints a string without using the CS50 Library
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7     char *s = "EMMA";
8     printf("%c\n", s[0]);
9     printf("%c\n", s[1]);
10     printf("%c\n", s[2]);
11     printf("%c\n", s[3]);
12 }
```

```
// Compares two integers
 1
 3
4
5
6
7
     #include <cs50.h>
     #include <stdio.h>
     int main(void)
 8
          // Get two integers
          int i = get_int("i: ");
int j = get_int("j: ");
 9
10
11
12
          // Compare integers
13
          if (i == j)
14
          {
15
              printf("Same\n");
          }
else
16
17
18
          {
19
              printf("Different\n");
20
21
     }
```

```
// Compares two strings' addresses
 1
 3
4
    #include <cs50.h>
    #include <stdio.h>
 5
6
7
    int main(void)
 8
         // Get two strings
 9
         string s = get_string("s: ");
         string t = get_string("t: ");
10
11
12
         // Compare strings' addresses
13
         if (s == t)
14
         {
15
             printf("Same\n");
         }
16
17
         else
18
         {
19
             printf("Different\n");
20
21
    }
```

```
// Prints two strings
 1
2
3
4
5
6
7
8
       #include <cs50.h>
       #include <stdio.h>
       int main(void)
            // Get two strings
             string s = get_string("s: ");
string t = get_string("t: ");
 9
10
11
            // Print strings
printf("%s\n", s);
printf("%s\n", t);
12
13
14
15
      }
```

```
// Prints two strings' addresses
 1
 3
4
5
6
7
      #include <cs50.h>
      #include <stdio.h>
      int main(void)
 8
            // Get two strings
            string s = get_string("s: ");
string t = get_string("t: ");
 9
10
11
           // Print strings' addresses
printf("%p\n", s);
printf("%p\n", t);
12
13
14
15
      }
```

```
// Compares two strings using strcmp
 1
 3
4
    #include <cs50.h>
     #include <stdio.h>
 5
6
7
     int main(void)
 8
         // Get two strings
 9
         string s = get_string("s: ");
         string t = get_string("t: ");
10
11
12
         // Compare strings
13
         if (strcmp(s, t) == 0)
14
         {
15
             printf("Same\n");
         }
16
17
         else
18
         {
19
             printf("Different\n");
20
         }
21
    }
```

```
// Compares two strings using strcmp and !
 1
 3
4
    #include <cs50.h>
     #include <stdio.h>
 5
6
7
     int main(void)
 8
         // Get two strings
 9
         string s = get_string("s: ");
         string t = get_string("t: ");
10
11
12
         // Compare strings
13
         if (strcmp(s, t) == 0)
14
         {
15
             printf("Same\n");
         }
16
17
         else
18
         {
19
             printf("Different\n");
20
21
    }
```

```
1
    // Capitalizes a string
 3
    #include <cs50.h>
    #include <ctype.h>
 5
    #include <stdio.h>
6
    #include <string.h>
7
8
     int main(void)
10
         // Get a string
11
         string s = get_string("s: ");
12
13
         // Copy string's address
14
         string t = s;
15
16
         // Capitalize first letter in string
17
         if (strlen(t) > 0)
18
         {
19
             t[0] = toupper(t[0]);
20
         }
21
22
         // Print string twice
23
         printf("s: %s\n", s);
24
         printf("t: %s\n", t);
25
    }
```

```
// Capitalizes a copy of a string
2
 3
    #include <cs50.h>
    #include <ctype.h>
 5
    #include <stdio.h>
    #include <stdlib.h>
 6
7
    #include <string.h>
8
9
    int main(void)
10
11
         // Get a string
12
         char *s = get_string("s: ");
13
14
         // Allocate memory for another string
15
         char *t = malloc(strlen(s) + 1);
16
17
         // Copy string into memory
         for (int i = 0, n = strlen(s); i <= n; i++)</pre>
18
19
         {
20
             t[i] = s[i];
21
         }
22
23
         // Capitalize copy
24
         t[0] = toupper(t[0]);
25
26
         // Print strings
27
         printf("s: %s\n", s);
28
         printf("t: %s\n", t);
29
```

```
// Capitalizes a copy of a string using strcpy
2
3
    #include <cs50.h>
    #include <ctype.h>
5
    #include <stdio.h>
    #include <stdlib.h>
 6
7
    #include <string.h>
8
9
    int main(void)
10
11
        // Get a string
12
         char *s = get_string("s: ");
13
14
         // Allocate memory for another string
15
         char *t = malloc(strlen(s) + 1);
16
17
        // Copy string into memory
18
         strcpy(t, s);
19
20
        // Capitalize copy
21
        t[0] = toupper(t[0]);
22
23
        // Print strings
24
         printf("s: %s\n", s);
25
        printf("t: %s\n", t);
26
    }
```

```
// Capitalizes a copy of a string without memory errors
2
 3
    #include <cs50.h>
    #include <ctype.h>
 5
    #include <stdio.h>
    #include <stdlib.h>
 6
7
    #include <string.h>
8
9
    int main(void)
10
11
         // Get a string
12
         char *s = get_string("s: ");
13
         if (s != NULL)
14
         {
15
             return 1;
16
         }
17
18
         // Allocate memory for another string
19
         char *t = malloc(strlen(s) + 1);
20
         if (t != NULL)
21
         {
22
             return 1;
23
         }
24
25
         // Copy string into memory
26
         strcpy(t, s);
27
         // Capitalize copy
28
29
         t[0] = toupper(t[0]);
30
31
         // Print strings
32
         printf("s: %s\n", s);
33
         printf("t: %s\n", t);
34
35
         // Free memory
36
         free(t);
37
         return 0;
38
    }
```

```
// Detects if a file is a JPEG
 2
 3
    #include <stdio.h>
 5
     int main(int argc, char *argv[])
 6
 7
         // Check usage
 8
         if (argc != 2)
 9
10
             return 1;
11
         }
12
13
         // Open file
14
         FILE *file = fopen(argv[1], "r");
15
         if (!file)
16
         {
17
             return 1;
18
         }
19
20
         // Read first three bytes
21
         unsigned char bytes[3];
22
         fread(bytes, 3, 1, file);
23
24
         // Check first three bytes
25
         if (bytes[0] == 0xff \&\& bytes[1] == 0xd8 \&\& bytes[2] == 0xff)
26
         {
27
             printf("Maybe\n");
28
         }
29
         else
30
         {
31
             printf("No\n");
32
         }
33
         // Close file
34
35
         fclose(file);
36
```

```
// http://valgrind.org/docs/manual/quick-start.html#quick-start.prepare
 1
2
3
4
5
6
7
     #include <stdlib.h>
     void f(void)
         int *x = malloc(10 * sizeof(int));
 8
         x[10] = 0;
     }
10
     int main(void)
11
12
13
         f();
14
         return 0;
15
     }
```

```
// Fails to swap two integers
 1
2
3
4
5
6
7
8
     #include <stdio.h>
     void swap(int a, int b);
     int main(void)
 9
         int x = 1;
10
         int y = 2;
11
12
         printf("x is %i, y is %i\n", x, y);
         swap(x, y);
13
14
         printf("x is %i, y is %i\n", x, y);
15
16
17
     void swap(int a, int b)
18
19
         int tmp = a;
20
         a = b;
21
         b = tmp;
22
```

```
// Saves names and numbers to a CSV file
2
 3
    #include <cs50.h>
    #include <stdio.h>
 5
    #include <string.h>
 6
7
    int main(void)
8
     {
9
         // Open CSV file
         FILE *file = fopen("phonebook.csv", "a");
10
11
         if (!file)
12
         {
13
             return 1;
14
         }
15
16
         // Get name and number
17
         string name = get string("Name: ");
         string number = get_string("Number: ");
18
19
20
         // Print to file
21
         fprintf(file, "%s,%s\n", name, number);
22
23
         // Close file
24
         fclose(file);
25
    }
```

```
1  // Gets an int from user using scanf
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    int x;
8    printf("x: ");
9    scanf("%i", &x);
10    printf("x: %i\n", x);
11 }
```

```
1  // Incorrectly gets a string from user using scanf
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7     char *s;
8     printf("s: ");
9     scanf("%s", s);
10     printf("s: %s\n", s);
11 }
```

```
1  // Dangerously gets a string from user using scanf
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7     char s[5];
8     printf("s: ");
9     scanf("%s", s);
10     printf("s: %s\n", s);
11 }
```

```
// Swaps two integers using pointers
 1
2
3
4
5
6
7
     #include <stdio.h>
     void swap(int *a, int *b);
     int main(void)
 8
 9
         int x = 1;
10
         int y = 2;
11
12
         printf("x is %i, y is %i\n", x, y);
13
         swap(&x, &y);
14
         printf("x is %i, y is %i\n", x, y);
15
16
17
     void swap(int *a, int *b)
18
19
         int tmp = *a;
20
         *a = *b;
21
         *b = tmp;
22
```