

```
1 // Implements linear search for integers
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // An array of integers
9     int numbers[] = {20, 500, 10, 5, 100, 1, 50};
10
11     // Search for number
12     int n = get_int("Number: ");
13     for (int i = 0; i < 7; i++)
14     {
15         if (numbers[i] == n)
16         {
17             printf("Found\n");
18             return 0;
19         }
20     }
21     printf("Not found\n");
22     return 1;
23 }
```

```
1 // Implements linear search for strings
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     // An array of strings
10    string strings[] = {"battleship", "boot", "cannon", "iron", "thimble", "top hat"};
11
12    // Search for string
13    string s = get_string("String: ");
14    for (int i = 0; i < 6; i++)
15    {
16        if (strcmp(strings[i], s) == 0)
17        {
18            printf("Found\n");
19            return 0;
20        }
21    }
22    printf("Not found\n");
23    return 1;
24 }
```

```
1 // Implements a phone book without structs
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     // Arrays of strings
10    string names[] = {"Carter", "David"};
11    string numbers[] = {"+1-617-495-1000", "+1-949-468-2750"};
12
13    // Search for name
14    string name = get_string("Name: ");
15    for (int i = 0; i < 2; i++)
16    {
17        if (strcmp(names[i], name) == 0)
18        {
19            printf("Found %s\n", numbers[i]);
20            return 0;
21        }
22    }
23    printf("Not found\n");
24    return 1;
25 }
```

```
1 // Implements a phone book with structs
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 typedef struct
8 {
9     string name;
10    string number;
11 }
12 person;
13
14 int main(void)
15 {
16     person people[2];
17
18     people[0].name = "Carter";
19     people[0].number = "+1-617-495-1000";
20
21     people[1].name = "David";
22     people[1].number = "+1-949-468-2750";
23
24     // Search for name
25     string name = get_string("Name: ");
26     for (int i = 0; i < 2; i++)
27     {
28         if (strcmp(people[i].name, name) == 0)
29         {
30             printf("Found %s\n", people[i].number);
31             return 0;
32         }
33     }
34     printf("Not found\n");
35     return 1;
36 }
```

```
1 // Draws a pyramid using iteration
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 void draw(int n);
7
8 int main(void)
9 {
10     // Get height of pyramid
11     int height = get_int("Height: ");
12
13     // Draw pyramid
14     draw(height);
15 }
16
17 void draw(int n)
18 {
19     // Draw pyramid of height n
20     for (int i = 0; i < n; i++)
21     {
22         for (int j = 0; j < i + 1; j++)
23         {
24             printf("#");
25         }
26         printf("\n");
27     }
28 }
```

```
1 // Draws a pyramid (incorrectly) using recursion,
2 // must be compiled with -Wno-infinite-recursion
3
4 #include <cs50.h>
5 #include <stdio.h>
6
7 void draw(int n);
8
9 int main(void)
10 {
11     draw(1);
12 }
13
14 void draw(int n)
15 {
16     for (int i = 0; i < n; i++)
17     {
18         printf("#");
19     }
20     printf("\n");
21
22     draw(n + 1);
23 }
```

```
1 // Draws a pyramid using recursion
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 void draw(int n);
7
8 int main(void)
9 {
10     // Get height of pyramid
11     int height = get_int("Height: ");
12
13     // Draw pyramid
14     draw(height);
15 }
16
17 void draw(int n)
18 {
19     // If nothing to draw
20     if (n <= 0)
21     {
22         return;
23     }
24
25     // Draw pyramid of height n - 1
26     draw(n - 1);
27
28     // Draw one more row of width n
29     for (int i = 0; i < n; i++)
30     {
31         printf("#");
32     }
33     printf("\n");
34 }
```