

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

```
1 // Implements linear search for names
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 int main(void)
8 {
9     // A n array of names
10    string names[] = {"B ill", "C harlie", "F red", "George", "Ginny", "P ercy", "R on"};
11
12    // S earch for R on
13    for (int i = 0; i < 7; i++)
14    {
15        if (strcmp(names[i], "R on") == 0)
16        {
17            printf("F ound\n");
18            return 0;
19        }
20    }
21    printf("N ot found\n");
22    return 1;
23 }
```

```
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     string names[] = {"Carter", "David"};
10    string numbers[] = {"+1-617-495-1000", "+1-949-468-2750"};
11
12    for (int i = 0; i < 2; i++)
13    {
14        if (strcmp(names[i], "David") == 0)
15        {
16            printf("Found %s\n", numbers[i]);
17            return 0;
18        }
19    }
20    printf("Not found\n");
21    return 1;
22 }
```

```
1 // Implements a phone b ook with structs
2
3 #include <cs50.h>
4 #include <stdio.h>
5 #include <string.h>
6
7 ty p ed e$ truc t
8 {
9     string name;
10    string numb er;
11 }
12 person;
13
14 int main(void)
15 {
16     person people[2] ;
17
18     people[0] name = "C arter";
19     people[0] numb er= "+1-617-495-1000";
20
21     people[1] name = "David";
22     people[1] numb er= "+1-949-468-2750";
23
24     // S earch for David
25     for (int i = 0; i < 2; i++)
26     {
27         if (strcmp(people[i] name, "David") == 0)
28         {
29             printf("F ound %\sn", people[i] numb er);
30             return 0;
31         }
32     }
33     printf("N ot found\n");
34     return 1;
35 }
```

```
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 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 }
```