```
1   #include <stdio.h>
2
3   int main(void)
4   {
5       printf("hello, world\n");
6   }
```

```c
1  #include <cs50.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      string name = get_string("What's your name? ");
7      printf("hello, %s\n", name);
8  }
```

```
1   // Buggy example for printf
2
3   #include <stdio.h>
4
5   int main(void)
6   {
7       for (int i = 0; i <= 3; i++)
8       {
9           printf("#\n");
10      }
11  }
```

```c
// Buggy example for printf

#include <stdio.h>

int main(void)
{
    for (int i = 0; i <= 3; i++)
    {
        printf("i is %i\n", i);
        printf("#\n");
    }
}
```

```c
1   // Buggy example for debug50
2
3   #include <cs50.h>
4   #include <stdio.h>
5
6   void print_column(int height);
7
8   int main(void)
9   {
10      int h = get_int("Height: ");
11      print_column(h);
12  }
13
14  void print_column(int height)
15  {
16      for (int i = 0; i <= height; i++)
17      {
18          printf("#\n");
19      }
20  }
```

```c
// Averages three (hardcoded) numbers

#include <stdio.h>

int main(void)
{
    // Scores
    int score1 = 72;
    int score2 = 73;
    int score3 = 33;

    // Print average
    printf("Average: %f\n", (score1 + score2 + score3) / 3.0);
}
```

```c
// Averages three (hardcoded) numbers using an array

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

int main(void)
{
    // Scores
    int scores[3];
    scores[0] = 72;
    scores[1] = 73;
    scores[2] = 33;

    // Print average
    printf("Average: %f\n", (scores[0] + scores[1] + scores[2]) / 3.0);
}
```

```c
// Averages three numbers using an array

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

int main(void)
{
    // Get scores
    int scores[3];
    scores[0] = get_int("Score: ");
    scores[1] = get_int("Score: ");
    scores[2] = get_int("Score: ");

    // Print average
    printf("Average: %f\n", (scores[0] + scores[1] + scores[2]) / 3.0);
}
```

```c
// Averages three numbers using an array and a loop

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

int main(void)
{
    // Get scores
    int scores[3];
    for (int i = 0; i < 3; i++)
    {
        scores[i] = get_int("Score: ");
    }

    // Print average
    printf("Average: %f\n", (scores[0] + scores[1] + scores[2]) / 3.0);
}
```

```
1   // Averages three numbers using an array, a constant, and a helper function
2
3   #include <cs50.h>
4   #include <stdio.h>
5
6   // Constant
7   const int N = 3;
8
9   // Prototype
10  float average(int length, int array[]);
11
12  int main(void)
13  {
14      // Get scores
15      int scores[N];
16      for (int i = 0; i < N; i++)
17      {
18          scores[i] = get_int("Score: ");
19      }
20
21      // Print average
22      printf("Average: %f\n", average(N, scores));
23  }
24
25  float average(int length, int array[])
26  {
27      // Calculate average
28      int sum = 0;
29      for (int i = 0; i < length; i++)
30      {
31          sum += array[i];
32      }
33      return sum / (float) length;
34  }
```

```c
// Prints chars

#include <stdio.h>

int main(void)
{
    char c1 = 'H';
    char c2 = 'I';
    char c3 = '!';

    printf("%c%c%c\n", c1, c2, c3);
}
```

```c
// Prints chars' ASCII codes

#include <stdio.h>

int main(void)
{
    char c1 = 'H';
    char c2 = 'I';
    char c3 = '!';

    printf("%i %i %i\n", c1, c2, c3);
}
```

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

```c
// Treats string as array

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

int main(void)
{
    string s = "HI!";
    printf("%c%c%c\n", s[0], s[1], s[2]);
}
```

```
 1   // Prints string's ASCII codes, including NUL
 2
 3   #include <cs50.h>
 4   #include <stdio.h>
 5
 6   int main(void)
 7   {
 8       string s = "HI!";
 9       printf("%i %i %i %i\n", s[0], s[1], s[2], s[3]);
10   }
```

```c
// Multiple strings

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

int main(void)
{
    string s = "HI!";
    string t = "BYE!";

    printf("%s\n", s);
    printf("%s\n", t);
}
```

```c
// Array of strings

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

int main(void)
{
    string words[2];

    words[0] = "HI!";
    words[1] = "BYE!";

    printf("%s\n", words[0]);
    printf("%s\n", words[1]);
}
```

```c
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    string words[2];

    words[0] = "HI!";
    words[1] = "BYE!";

    printf("%c%c%c\n", words[0][0], words[0][1], words[0][2]);
    printf("%c%c%c%c\n", words[1][0], words[1][1], words[1][2], words[1][3]);
}
```

```c
// Determines the length of a string

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

int main(void)
{
    // Prompt for user's name
    string name = get_string("Name: ");

    // Count number of characters up until '\0' (aka NUL)
    int n = 0;
    while (name[n] != '\0')
    {
        n++;
    }
    printf("%i\n", n);
}
```

```c
1    // Determines the length of a string using a function
2
3    #include <cs50.h>
4    #include <stdio.h>
5
6    int string_length(string s);
7
8    int main(void)
9    {
10       // Prompt for user's name
11       string name = get_string("Name: ");
12       int length = string_length(name);
13       printf("%i\n", length);
14   }
15
16   int string_length(string s)
17   {
18       // Count number of characters up until '\0' (aka NUL)
19       int n = 0;
20       while (s[n] != '\0')
21       {
22           n++;
23       }
24       return n;
25   }
```

```c
 1    // Determines the length of a string using a function
 2
 3    #include <cs50.h>
 4    #include <stdio.h>
 5    #include <string.h>
 6
 7    int main(void)
 8    {
 9        // Prompt for user's name
10        string name = get_string("Name: ");
11        int length = strlen(name);
12        printf("%i\n", length);
13    }
```

```
1    // Prints string char by char, using strlen
2
3    #include <cs50.h>
4    #include <stdio.h>
5    #include <string.h>
6
7    int main(void)
8    {
9        string s = get_string("Input:  ");
10       printf("Output: ");
11       for (int i = 0; i < strlen(s); i++)
12       {
13           printf("%c", s[i]);
14       }
15       printf("\n");
16   }
```

```c
// Prints string char by char, using strlen, remembering string's length

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

int main(void)
{
    string s = get_string("Input:  ");
    printf("Output: ");
    for (int i = 0, n = strlen(s); i < n; i++)
    {
        printf("%c", s[i]);
    }
    printf("\n");
}
```

```
1    // Uppercases a string
2
3    #include <cs50.h>
4    #include <stdio.h>
5    #include <string.h>
6
7    int main(void)
8    {
9        string s = get_string("Before: ");
10       printf("After:  ");
11       for (int i = 0, n = strlen(s); i < n; i++)
12       {
13           if (s[i] >= 'a' && s[i] <= 'z')
14           {
15               printf("%c", s[i] - 32);
16           }
17           else
18           {
19               printf("%c", s[i]);
20           }
21       }
22       printf("\n");
23   }
```

```c
1    // Uppercases string using ctype library (and an unnecessary condition)
2
3    #include <cs50.h>
4    #include <ctype.h>
5    #include <stdio.h>
6    #include <string.h>
7
8    int main(void)
9    {
10       string s = get_string("Before: ");
11       printf("After:  ");
12       for (int i = 0, n = strlen(s); i < n; i++)
13       {
14           if (islower(s[i]))
15           {
16               printf("%c", toupper(s[i]));
17           }
18           else
19           {
20               printf("%c", s[i]);
21           }
22       }
23       printf("\n");
24   }
```

```
1    // Uppercases string using ctype library
2
3    #include <cs50.h>
4    #include <ctype.h>
5    #include <stdio.h>
6    #include <string.h>
7
8    int main(void)
9    {
10       string s = get_string("Before: ");
11       printf("After:  ");
12       for (int i = 0, n = strlen(s); i < n; i++)
13       {
14           printf("%c", toupper(s[i]));
15       }
16       printf("\n");
17   }
```

```
1   // Uses get_string
2
3   #include <cs50.h>
4   #include <stdio.h>
5
6   int main(void)
7   {
8       string answer = get_string("What's your name? ");
9       printf("hello, %s\n", answer);
10  }
```

```c
// Uses get_string with better-named variable

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

int main(void)
{
    string name = get_string("What's your name? ");
    printf("hello, %s\n", name);
}
```

```
1    // Prints a command-line argument
2
3    #include <cs50.h>
4    #include <stdio.h>
5
6    int main(int argc, string argv[])
7    {
8        printf("hello, %s\n", argv[1]);
9    }
```

```c
// Prints a command-line argument

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

int main(int argc, string argv[])
{
    if (argc == 2)
    {
        printf("hello, %s\n", argv[1]);
    }
    else
    {
        printf("hello, world\n");
    }
}
```

```c
// Prints command-line arguments

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

int main(int argc, string argv[])
{
    for (int i = 0; i < argc; i++)
    {
        printf("%s\n", argv[i]);
    }
}
```

```c
// Returns explicit value from main

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

int main(int argc, string argv[])
{
    if (argc != 2)
    {
        printf("Missing command-line argument\n");
        return 1;
    }
    printf("hello, %s\n", argv[1]);
    return 0;
}
```