```c
// Buggy example for printf

#include <stdio.h>

int main(void)
{
    for (int i = 0; i <= 10; i++)
    {
        printf("#\n");
    }
}
```

```c
// Buggy example for debug50

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

int get_negative_int(void);

int main(void)
{
    int i = get_negative_int();
    printf("%i\n", i);
}

// Prompt user for positive integer
int get_negative_int(void)
{
    int n;
    do
    {
        n = get_int("Negative Integer: ");
    }
    while (n < 0);
    return n;
}
```

scores0.c

```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);
}
```

scores1.c

```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);
}
```

scores2.c

```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);
}
```

scores4.c

```c
// Averages three numbers using an array, a constant, and a helper function

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

// Constant
const int N = 3;

// Prototype
float average(int length, int array[]);

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

    // Print average
    printf("Average: %f\n", average(N, scores));
}

float average(int length, int array[])
{
    // Calculate average
    int sum = 0;
    for (int i = 0; i < length; i++)
    {
        sum += array[i];
    }
    return sum / (float) length;
}
```

brick0.c

```c
// Prints char

#include <stdio.h>

int main(void)
{
    char c = '#';

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

brick1.c

```c
// Prints char's ASCII code

#include <stdio.h>

int main(void)
{
    char c = '#';

    printf("%i\n", c);
}
```

hi0.c

```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);
}
```

```c
// Prints string

#include <stdio.h>

int main(void)
{
    string s = "HI!";
    printf("%s\n", s);
}
```

hi3.c

```c
// Prints string's ASCII codes

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

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

hi4.c

```c
// Prints string's ASCII codes, including NUL

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

int main(void)
{
    string s = "HI!";
    printf("%i %i %i %i\n", s[0], s[1], s[2], s[3]);
}
```

string0.c

```c
// Prints string char by char

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

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

string1.c

```c
// Prints string char by char, using strlen

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

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

string2.c

```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");
}
```

length.c

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

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

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

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

ascii0.c

```c
// Explicitly casts chars to ints

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

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

ascii1.c

```c
// Implicitly casts chars to ints

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

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

uppercase0.c

```c
// Uppercases a string

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

int main(void)
{
    string s = get_string("Before: ");
    printf("After:  ");
    for (int i = 0, n = strlen(s); i < n; i++)
    {
        if (s[i] >= 'a' && s[i] <= 'z')
        {
            printf("%c", s[i] - 32);
        }
        else
        {
            printf("%c", s[i]);
        }
    }
    printf("\n");
}
```

uppercase1.c

```c
// Uppercases string using ctype library (and an unnecessary condition)

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

int main(void)
{
    string s = get_string("Before: ");
    printf("After:  ");
    for (int i = 0, n = strlen(s); i < n; i++)
    {
        if (islower(s[i]))
        {
            printf("%c", toupper(s[i]));
        }
        else
        {
            printf("%c", s[i]);
        }
    }
    printf("\n");
}
```

```c
// Uppercases string using ctype library

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

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

argv0.c

```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");
    }
}
```

argv1.c

```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]);
    }
}
```

argv2.c

```c
// Prints characters in an array of strings

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

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

exit.c

```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;
}
```