```
1   // A program that says hello to the world
2
3   #include <stdio.h>
4
5   int main(void)
6   {
7       printf("hello, world\n");
8   }
```

```
1    // get_string and printf with incorrect placeholder
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, answer\n");
10   }
```

```
1   // get_string and printf with %s
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
// Conditional, Boolean expression, relational operator

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

int main(void)
{
    // Prompt user for integers
    int x = get_int("What's x? ");
    int y = get_int("What's y? ");

    // Compare integers
    if (x < y)
    {
        printf("x is less than y\n");
    }
}
```

```c
// Conditionals that are mutually exclusive

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

int main(void)
{
    // Prompt user for integers
    int x = get_int("What's x? ");
    int y = get_int("What's y? ");

    // Compare integers
    if (x < y)
    {
        printf("x is less than y\n");
    }
    else
    {
        printf("x is not less than y\n");
    }
}
```

```c
// Conditionals that aren't mutually exclusive

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

int main(void)
{
    // Prompt user for integers
    int x = get_int("What's x? ");
    int y = get_int("What's y? ");

    // Compare integers
    if (x < y)
    {
        printf("x is less than y\n");
    }
    if (x > y)
    {
        printf("x is greater than y\n");
    }
    if (x == y)
    {
        printf("x is equal to y\n");
    }
}
```

```c
// Conditional that isn't necessary

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

int main(void)
{
    // Prompt user for integers
    int x = get_int("What's x? ");
    int y = get_int("What's y? ");

    // Compare integers
    if (x < y)
    {
        printf("x is less than y\n");
    }
    else if (x > y)
    {
        printf("x is greater than y\n");
    }
    else if (x == y)
    {
        printf("x is equal to y\n");
    }
}
```

```c
// Comparing against lowercase char

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

int main(void)
{
    // Prompt user to agree
    char c = get_char("Do you agree? ");

    // Check whether agreed
    if (c == 'y')
    {
        printf("Agreed.\n");
    }
    else if (c == 'n')
    {
        printf("Not agreed.\n");
    }
}
```

```c
// Comparing against lowercase and uppercase char

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

int main(void)
{
    // Prompt user to agree
    char c = get_char("Do you agree? ");

    // Check whether agreed
    if (c == 'y')
    {
        printf("Agreed.\n");
    }
    else if (c == 'Y')
    {
        printf("Agreed.\n");
    }
    else if (c == 'n')
    {
        printf("Not agreed.\n");
    }
    else if (c == 'N')
    {
        printf("Not agreed.\n");
    }
}
```

```c
// Logical operators

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

int main(void)
{
    // Prompt user to agree
    char c = get_char("Do you agree? ");

    // Check whether agreed
    if (c == 'Y' || c == 'y')
    {
        printf("Agreed.\n");
    }
    else if (c == 'N' || c == 'n')
    {
        printf("Not agreed.\n");
    }
}
```

```c
// Opportunity for better design

#include <stdio.h>

int main(void)
{
    printf("meow\n");
    printf("meow\n");
    printf("meow\n");
}
```

```c
// Better design

#include <stdio.h>

int main(void)
{
    int i = 3;
    while (i > 3)
    {
        printf("meow\n");
        i--;
    }
}
```

```c
// Better design

#include <stdio.h>

int main(void)
{
    int i = 3;
    while (i > 3)
    {
        printf("%i\n", i);
        i--;
    }
}
```

```c
// Better design

#include <stdio.h>

int main(void)
{
    int i = 0;
    while (i < 3)
    {
        printf("meow\n");
        i++;
    }
}
```

```c
// Better design

#include <stdio.h>

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

```c
// Abstraction

#include <stdio.h>

void meow(void);

int main(void)
{
    for (int i = 0; i < 3; i++)
    {
        meow();
    }
}

// Meow once
void meow(void)
{
    printf("meow\n");
}
```

```c
// Abstraction with parameterization

#include <stdio.h>

void meow(int n);

int main(void)
{
    meow(3);
}

// Meow some number of times
void meow(int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("meow\n");
    }
}
```

```
1   // Addition with int
2
3   #include <cs50.h>
4   #include <stdio.h>
5
6   int main(void)
7   {
8       // Prompt user for x
9       int x = get_int("x: ");
10
11      // Prompt user for y
12      int y = get_int("y: ");
13
14      // Perform addition
15      printf("%i\n", x + y);
16  }
```

```c
// Scope error

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

int add(void);

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Perform addition
    int z = add();
    printf("%i\n", z);
}

int add(void)
{
    return x + y;
}
```

```c
// Helper function with arguments and return value

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

int add(int a, int b);

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Perform addition
    int z = add(x, y);
    printf("%i\n", z);
}

int add(int a, int b)
{
    int c = a + b;
    return c;
}
```

```c
// Eliminates unnecessary variables

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

int add(int a, int b);

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Perform addition
    printf("%i\n", add(x, y));
}

int add(int a, int b)
{
    return a + b;
}
```

```c
// Addition with long

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

int main(void)
{
    // Prompt user for x
    long x = get_long("x: ");

    // Prompt user for y
    long y = get_long("y: ");

    // Perform addition
    printf("%li\n", x + y);
}
```

```c
// Prints a row of 4 question marks

#include <stdio.h>

int main(void)
{
    printf("????\n");
}
```

```c
// Prints a row of 4 question marks with a loop

#include <stdio.h>

int main(void)
{
    for (int i = 0; i < 4; i++)
    {
        printf("?");
    }
    printf("\n");
}
```

```c
// Prints a column of 3 bricks with a loop

#include <stdio.h>

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

```c
1   // Prints a 3-by-3 grid of bricks with nested loops
2
3   #include <stdio.h>
4
5   int main(void)
6   {
7       for (int i = 0; i < 3; i++)
8       {
9           for (int j = 0; j < 3; j++)
10          {
11              printf("#");
12          }
13          printf("\n");
14      }
15  }
```

```c
// Prints a 3-by-3 grid of bricks with nested loops using a constant

#include <stdio.h>

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

```c
// Prints an n-by-n grid of bricks with nested loops

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

int main(void)
{
    int n = get_int("Size: ");

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

```
1   // Prints an n-by-n grid of bricks, re-prompting user for positive integer
2
3   #include <cs50.h>
4   #include <stdio.h>
5
6   int main(void)
7   {
8       int n = get_int("Size: ");
9       while (n < 1)
10      {
11          n = get_int("Size: ");
12      }
13
14      for (int i = 0; i < n; i++)
15      {
16          for (int j = 0; j < n; j++)
17          {
18              printf("#");
19          }
20          printf("\n");
21      }
22  }
```

```c
// Prints an n-by-n grid of bricks, re-prompting user for positive integer

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

int main(void)
{
    // Get size of grid
    int n;
    do
    {
        n = get_int("Size: ");
    }
    while (n < 1);

    // Print grid of bricks
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            printf("#");
        }
        printf("\n");
    }
}
```

```c
// Division with ints, demonstrating truncation

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

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Divide x by y
    printf("%i\n", x / y);
}
```

```
1    // Uses %f
2
3    #include <cs50.h>
4    #include <stdio.h>
5
6    int main(void)
7    {
8        // Prompt user for x
9        int x = get_int("x: ");
10
11       // Prompt user for y
12       int y = get_int("y: ");
13
14       // Divide x by y
15       printf("%f\n", x / y);
16   }
```

```c
// Uses float variable

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

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Divide x by y
    float z = x / y;
    printf("%f\n", z);
}
```

```c
// Type casting

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

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Divide x by y
    float z = (float) x / (float) y;
    printf("%f\n", z);
}
```

```c
// Floating-point imprecision

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

int main(void)
{
    // Prompt user for x
    int x = get_int("x: ");

    // Prompt user for y
    int y = get_int("y: ");

    // Divide x by y
    float z = (float) x / (float) y;
    printf("%.20f\n", z);
}
```

```c
// Division with longs, demonstrating double

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

int main(void)
{
    // Prompt user for x
    long x = get_long("x: ");

    // Prompt user for y
    long y = get_long("y: ");

    // Divide x by y
    double z = (double) x / (double) y;
    printf("%.20f\n", z);
}
```