```
1
     // Logical operators
 3
    #include <cs50.h>
 4
    #include <stdio.h>
 5
 6
7
     int main(void)
 8
         // Prompt user to agree
 9
         char c = get char("Do you agree?\n");
10
         // Check whether agreed
11
12
         if (c == 'Y' || c == 'y')
13
14
             printf("Agreed.\n");
15
16
         else if (c == 'N' || c == 'n')
17
             printf("Not agreed.\n");
18
19
         }
20
     }
```

```
// Conditions and relational operators
 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
         // Prompt user for y
11
         int y = get_int("y: ");
12
13
14
         // Compare x and y
15
         if (x < y)
16
         {
17
             printf("x is less than y\n");
18
19
         else if (x > y)
20
             printf("x is greater than y\n");
21
22
         }
23
         else
24
         {
             printf("x is equal to y\n");
25
26
         }
27
     }
```

```
// Opportunity for better design

#include <stdio.h>

int main(void)

frintf("cough\n");
printf("cough\n");
printf("cough\n");
}
```

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

```
// Abstraction
 1
2
3
4
5
6
7
8
9
     #include <stdio.h>
     void cough(void);
     int main(void)
          for (int i = 0; i < 3; i++)
10
               cough();
11
12
13
14
     // Cough once
void cough(void)
15
16
17
          printf("cough\n");
18
19
```

```
// Abstraction with parameterization
 1
2
3
4
5
6
7
8
9
     #include <stdio.h>
     void cough(int n);
     int main(void)
     {
          cough(3);
10
11
12
     // Cough some number of times
13
     void cough(int n)
14
15
         for (int i = 0; i < n; i++)</pre>
16
17
              printf("cough\n");
18
19
     }
```

```
// Floating-point arithmetic with double
 1
2
3
4
5
6
7
     #include <cs50.h>
     #include <stdio.h>
     int main(void)
 8
         // Prompt user for x
 9
         double x = get_double("x: ");
10
         // Prompt user for y
11
         double y = get_double("y: ");
12
13
14
         // Perform division
15
         printf("x / y = %.50f\n", x / y);
16
    }
```

```
1  // get_float and printf with %f
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8    float price = get_float("What's the price?\n$");
9    printf("Your total is $%.2f.\n", price * 1.0625);
10 }
```

```
// Floating-point arithmetic with float
 1
2
3
4
5
6
7
     #include <cs50.h>
     #include <stdio.h>
     int main(void)
 8
         // Prompt user for x
         float x = get_float("x: ");
 9
10
         // Prompt user for y
11
         float y = get_float("y: ");
12
13
14
         // Perform division
15
         printf("x / y = %.50f\n", x / y);
16
    }
```

```
// A program that says hello to the world

#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

```
1  // get_int and printf with %i
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8    int age = get_int("What's your age?\n");
9    printf("You are at least %i days old.\n", age * 365);
10 }
```

```
1  // Prints a row of 4 question marks
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    printf("????\n");
8  }
```

```
1  // Prints a row of 4 question marks with a loop
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    for (int i = 0; i < 4; i++)
8     {
9       printf("?");
10    }
11    printf("\n");
12 }</pre>
```

```
// Prints a row of n question marks with a loop
 1
 3
4
    #include <cs50.h>
    #include <stdio.h>
 5
 6
7
     int main(void)
 8
         int n;
 9
         do
10
             n = get_int("Width: ");
11
12
13
         while (n < 1);
14
         for (int i = 0; i < n; i++)
15
16
             printf("?");
17
         printf("\n");
18
19
    }
```

```
1  // Prints a column of 3 bricks
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    printf("#\n");
8    printf("#\n");
9    printf("#\n");
10 }
```

```
1  // Prints a column of 3 bricks with a loop
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 }</pre>
```

```
// Prints a column of n bricks with a loop
 1
 3
4
    #include <cs50.h>
    #include <stdio.h>
 5
 6
7
     int main(void)
 8
         int n;
 9
         do
10
             n = get_int("Height: ");
11
12
13
        while (n < 1);
14
         for (int i = 0; i < n; i++)
15
16
             printf("#\n");
17
         }
18
     }
```

```
1  // Prints 3-by-3 grid of bricks
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7    printf("###\n");
8    printf("###\n");
9    printf("###\n");
10 }
```

```
1  // Prints a 3-by-3 grid of bricks with a loop
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 }</pre>
```

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

```
// Integer overflow
 1
2
3
4
5
6
7
8
     #include <stdio.h>
     #include <unistd.h>
     int main(void)
          // Iteratively double i
 9
          for (int i = 1; ; i *= 2)
10
              printf("%i\n", i);
sleep(1);
11
12
13
          }
14
     }
```

```
// Calculates a remainder
1
3
4
    #include <cs50.h>
    #include <stdio.h>
 5
6
    int main(void)
8
        // Prompt user for integer
9
         int n = get_int("n: ");
10
        // Check parity of integer
11
        if (n % 2 == 0)
12
13
14
             printf("even\n");
15
         }
        else
16
17
         {
             printf("odd\n");
18
19
         }
20
     }
```

```
// Abstraction and scope
 2
 3
    #include <cs50.h>
     #include <stdio.h>
 5
6
7
     int get_positive_int(void);
8
     int main(void)
10
         int i = get_positive_int();
         printf("%i\n", i);
11
     }
12
13
14
    // Prompt user for positive integer
    int get_positive_int(void)
15
16
     {
17
         int n;
18
         do
19
         {
20
             n = get_int("Positive Integer: ");
21
22
         while (n < 1);
23
         return n;
24
    }
```

```
1  // Math library
2
3  #include <cs50.h>
4  #include <math.h>
5  #include <stdio.h>
6
7  int main(void)
8  {
9     double base = get_double("Base: ");
10     double exponent = get_double("Exponent: ");
11     printf("Output: %.0f\n", pow(base, exponent));
12 }
```

```
// Return value
1
2
3
4
5
6
7
8
9
     #include <cs50.h>
     #include <stdio.h>
      int square(int n);
      int main(void)
          int input = get_int("Input: ");
printf("Output: %i\n", square(n));
11
      }
12
13
14
     // Square n
15
     int square(int n)
16
     {
17
           return n * n;
18
```

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