
```
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 %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", s);
10 }
```

```
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 }
```

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

```
1 // Truncation
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Get numbers from user
9     int x = get_int("x: ");
10    int y = get_int("y: ");
11
12    // Divide x by y
13    float z = x / y;
14
15    // Perform division
16    printf("%f\n", z);
17 }
```

```
1 // 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
11     // Prompt user for y
12     int y = get_int("y: ");
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     {
21         printf("x is greater than y\n");
22     }
23     else
24     {
25         printf("x is equal to y\n");
26     }
27 }
```

```
1 // Logical operators
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user to agree
9     char c = get_char("Do you agree? ");
10
11     // Check whether agreed
12     if (c == 'Y' || c == 'y')
13     {
14         printf("Agreed.\n");
15     }
16     else if (c == 'N' || c == 'n')
17     {
18         printf("Not agreed.\n");
19     }
20 }
```

```
1 // Opportunity for better design
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("meow\n");
8     printf("meow\n");
9     printf("meow\n");
10 }
```

```
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("meow\n");
10    }
11 }
```

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

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

```
1 // Abstraction and scope
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int get_positive_int(void);
7
8 int main(void)
9 {
10     int i = get_positive_int();
11     printf("%i\n", i);
12 }
13
14 // Prompt user for positive integer
15 int get_positive_int(void)
16 {
17     int n;
18     do
19     {
20         n = get_int("Positive Integer: ");
21     }
22     while (n < 1);
23     return n;
24 }
```

```
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 }
```

```
1 // Prints a row of n question marks with a loop
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Get positive integer from user
9     int n;
10    do
11    {
12        n = get_int("Width: ");
13    }
14    while (n < 1);
15
16    // // Print out that many question marks
17    for (int i = 0; i < n; i++)
18    {
19        printf("?");
20    }
21    printf("\n");
22 }
```

```
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 nested loops
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     for (int i = 0; i < 3; i++)
9     {
10        for (int j = 0; j < 3; j++)
11        {
12            printf("#");
13        }
14        printf("\n");
15    }
16 }
```

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

```
1 // Integer overflow
2
3 #include <stdbool.h>
4 #include <stdio.h>
5 #include <unistd.h>
6
7 int main(void)
8 {
9     // Iteratively double i
10    int i = 1;
11    while (true)
12    {
13        printf("%i\n", i);
14        sleep(1);
15        i *= 2;
16    }
17 }
```