CS1010

https://github.com/DigiPie/cs1010_tut_c09

Today’s plan

- Tutorial Segment
  - *Kahoot!*
  - *Discussion of problem sets*
- Consultation

Kahoot!

Problem Sets

10.1, 10.2, 11.1, 11.2, 11.3, 12.1

Problem Set 10.1 (a)

Negate the following logical expression, then apply De Morgan's Law to simplify the resulting expression.

(a) (x > 1) && (y != 10)

1. !((x > 1) && (y != 10))
2. !(x > 1) || !(y != 10)
3. (x <= 1) || (y == 10)

Problem Set 10.1 (b)

Negate the following logical expression, then apply De Morgan's Law to simplify the resulting expression.

(b) !eating && drinking

1. !(!eating && drinking)
2. eating) | | !drinking

Problem Set 10.1 (c)

Negate the following logical expression, then apply De Morgan's Law to simplify the resulting expression.

(c) (has_cs2030 || has_cs2113) && has_cs2040c

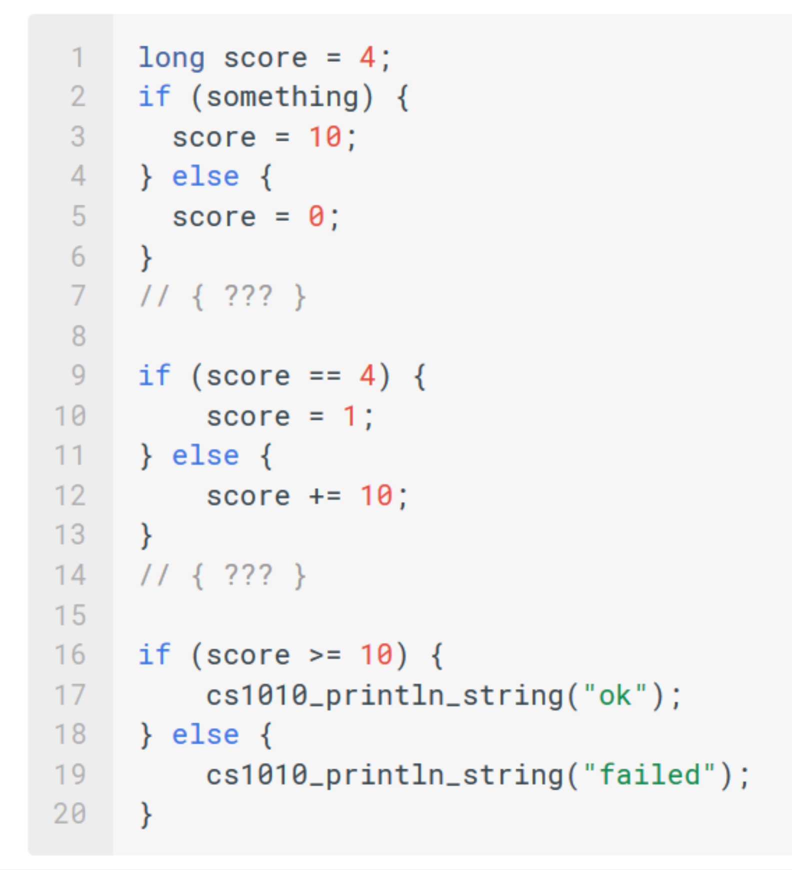
1. !((has_cs2030 || has_cs2113) && has_cs2040c)
2. !(has_cs2030 || has_cs2113) || !has_cs2040c
3. (!has_cs2030 && !has_cs2113) || !has_cs2040c

PROBLEM SETS

10.1, 10.2, 11.1, 11.2, 11.3, 12.1

Problem Set 10.2

In the code below, replace ??? with the appropriate assertion.



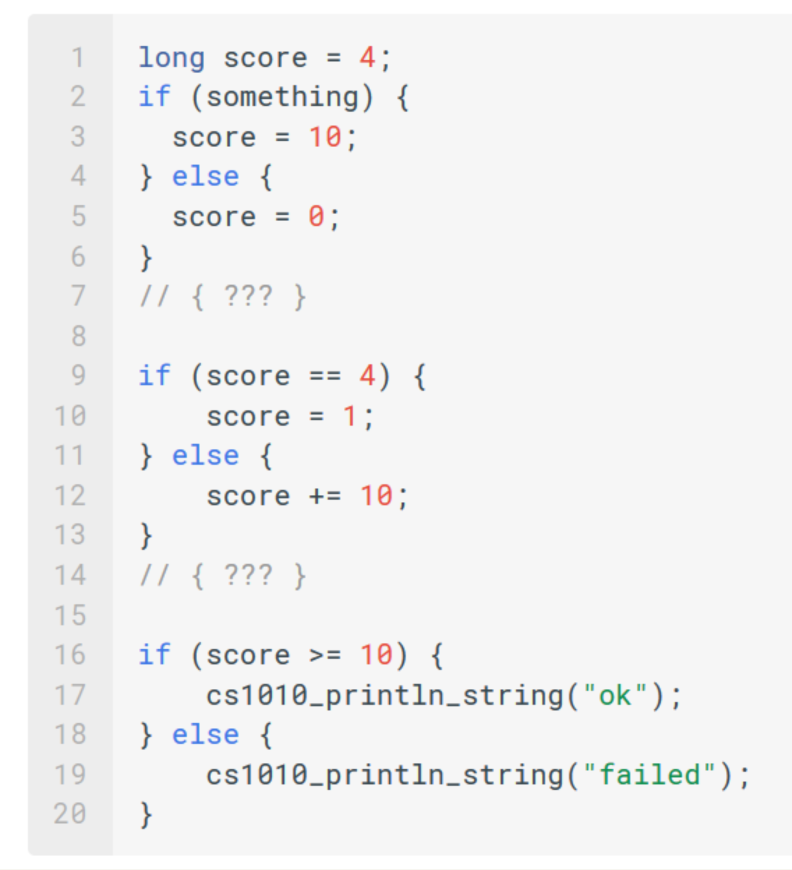
```
 1  long score = 4;
 2  if (something) {
 3    score = 10;
 4  } else {
 5    score = 0;
 6  }
 7  // { ??? }
 8
 9  if (score == 4) {
10      score = 1;
11  } else {
12      score += 10;
13  }
14  // { ??? }
15
16  if (score >= 10) {
17      cs1010_println_string("ok");
18  } else {
19      cs1010_println_string("failed");
20  }
```

7 // { score == ... || score == ... }

14 // { ... }

Problem Set 10.2

In the code below, replace ??? with the appropriate assertion.



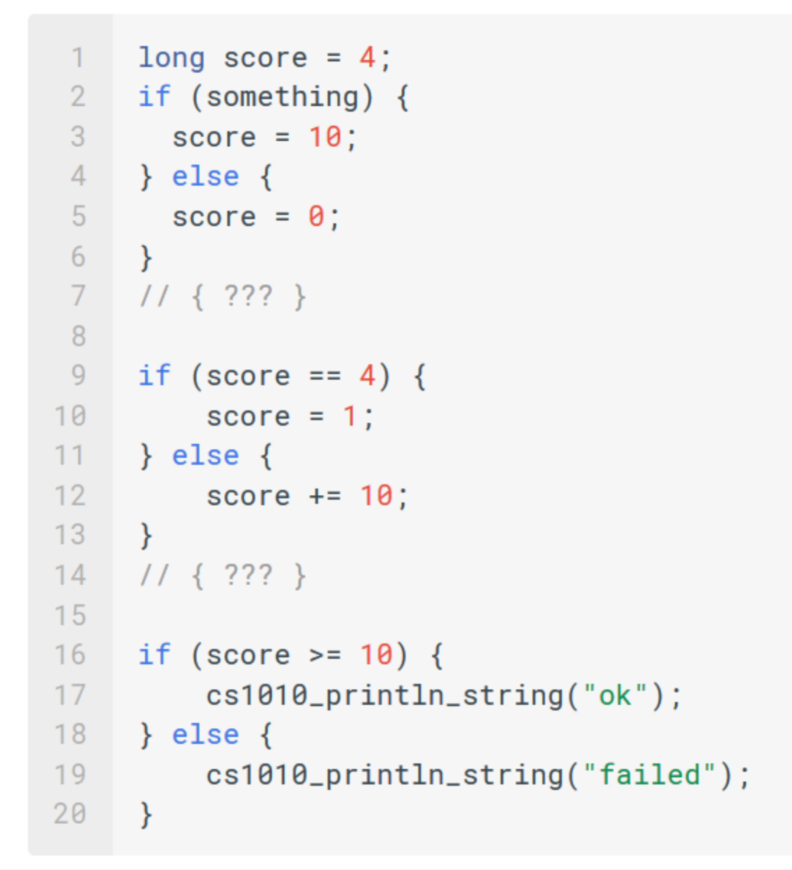
```
 1  long score = 4;
 2  if (something) {
 3    score = 10;
 4  } else {
 5    score = 0;
 6  }
 7  // { ??? }
 8
 9  if (score == 4) {
10      score = 1;
11  } else {
12      score += 10;
13  }
14  // { ??? }
15
16  if (score >= 10) {
17      cs1010_println_string("ok");
18  } else {
19      cs1010_println_string("failed");
20  }
```

7 // { score == 10 | | score == 0 }

14 // { score == ... | | score == ... }

Problem Set 10.2

In the code below, replace ??? with the appropriate assertion.



```
 1  long score = 4;
 2  if (something) {
 3    score = 10;
 4  } else {
 5    score = 0;
 6  }
 7  // { ??? }
 8
 9  if (score == 4) {
10      score = 1;
11  } else {
12      score += 10;
13  }
14  // { ??? }
15
16  if (score >= 10) {
17      cs1010_println_string("ok");
18  } else {
19      cs1010_println_string("failed");
20  }
```

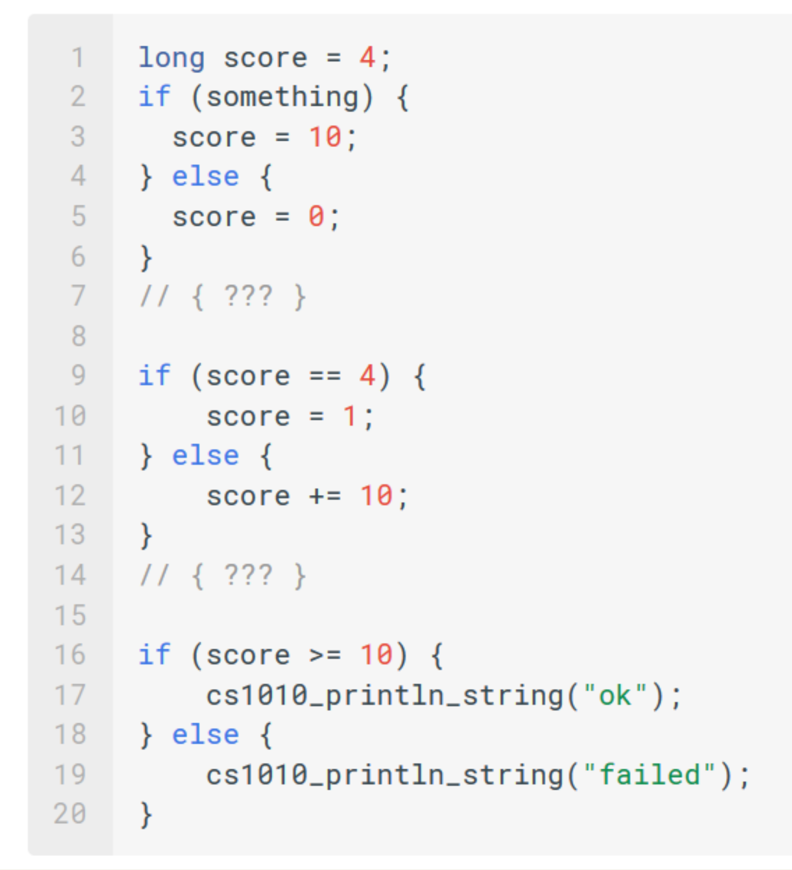
7 // { score == 10 | | score == 0 }

14 // { score == 20 | | score == 10 }

Problem Set 10.2

In the code below, replace ??? with the appropriate assertion.

What will be printed?



```
 1  long score = 4;
 2  if (something) {
 3    score = 10;
 4  } else {
 5    score = 0;
 6  }
 7  // { ??? }
 8
 9  if (score == 4) {
10      score = 1;
11  } else {
12      score += 10;
13  }
14  // { ??? }
15
16  if (score >= 10) {
17      cs1010_println_string("ok");
18  } else {
19      cs1010_println_string("failed");
20  }
```

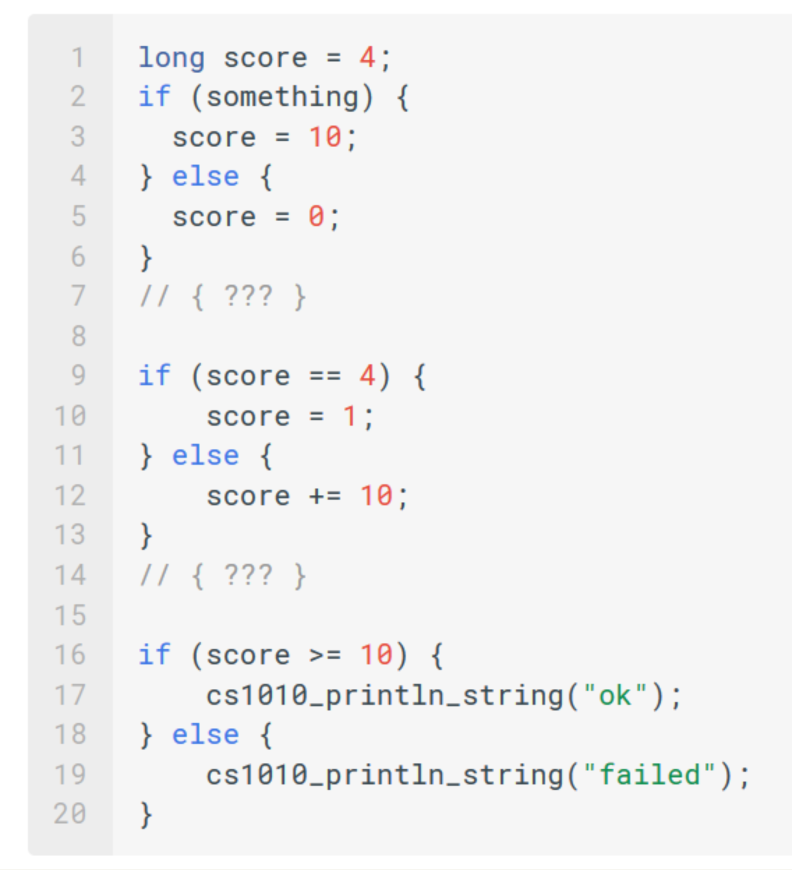
7 // { score == 10 || score == 0 }

14 // { score == 20 || score == 10 }

Problem Set 10.2

In the code below, replace ??? with the appropriate assertion.

What will be printed?



```
 1  long score = 4;
 2  if (something) {
 3    score = 10;
 4  } else {
 5    score = 0;
 6  }
 7  // { ??? }
 8
 9  if (score == 4) {
10      score = 1;
11  } else {
12      score += 10;
13  }
14  // { ??? }
15
16  if (score >= 10) {
17      cs1010_println_string("ok");
18  } else {
19      cs1010_println_string("failed");
20  }
```

7 // { score == 10 | | score == 0 }

14 // { score == 20 | | score == 10 }

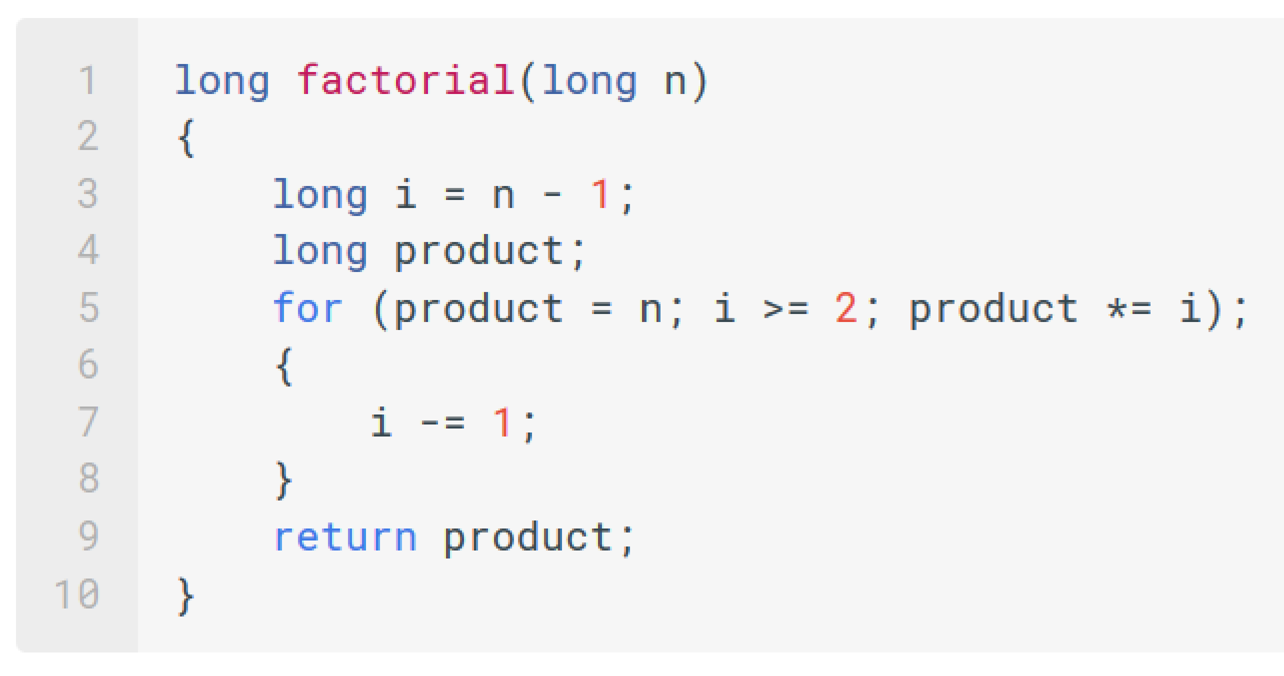
“ok” will always be printed.

PROBLEM SETS

10.1, 10.2, 11.1, 11.2, 11.3, 12.1

Problem Set 11.1

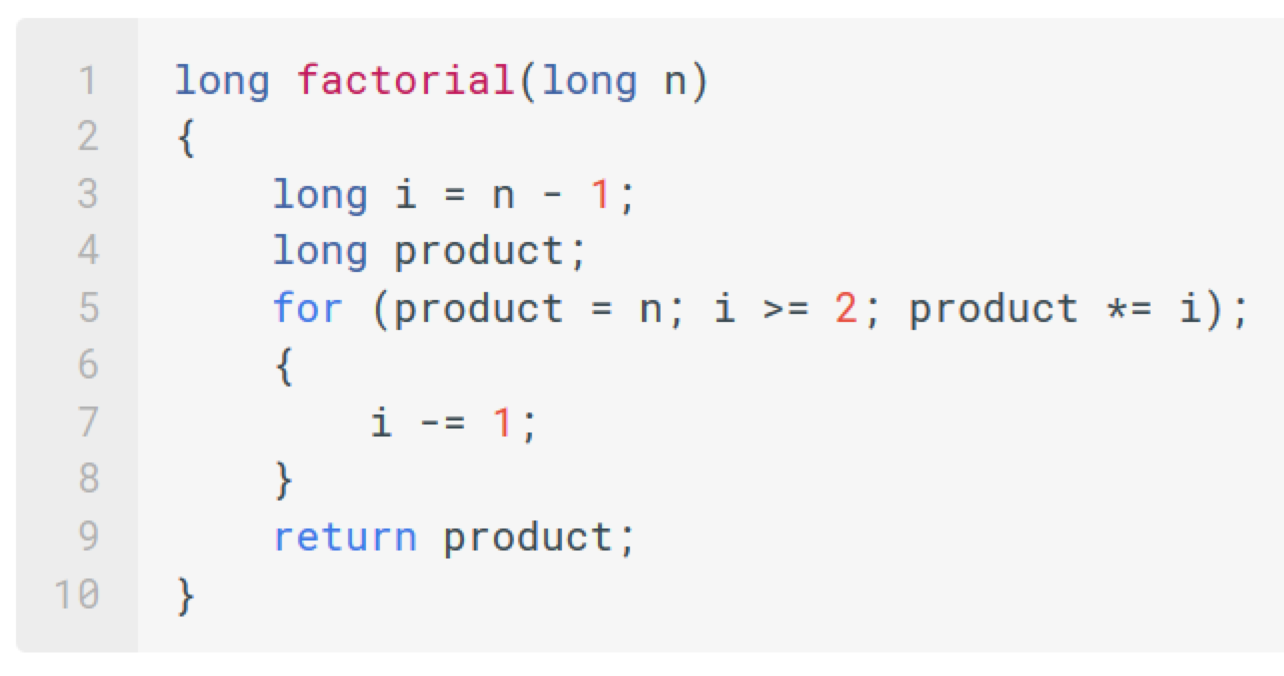
Does this code runs correctly?



```
long factorial(long n)
{
    long i = n - 1;
    long product;
    for (product = n; i >= 2; product *= i);
    {
        i -= 1;
    }
    return product;
}
```

Problem Set 11.1

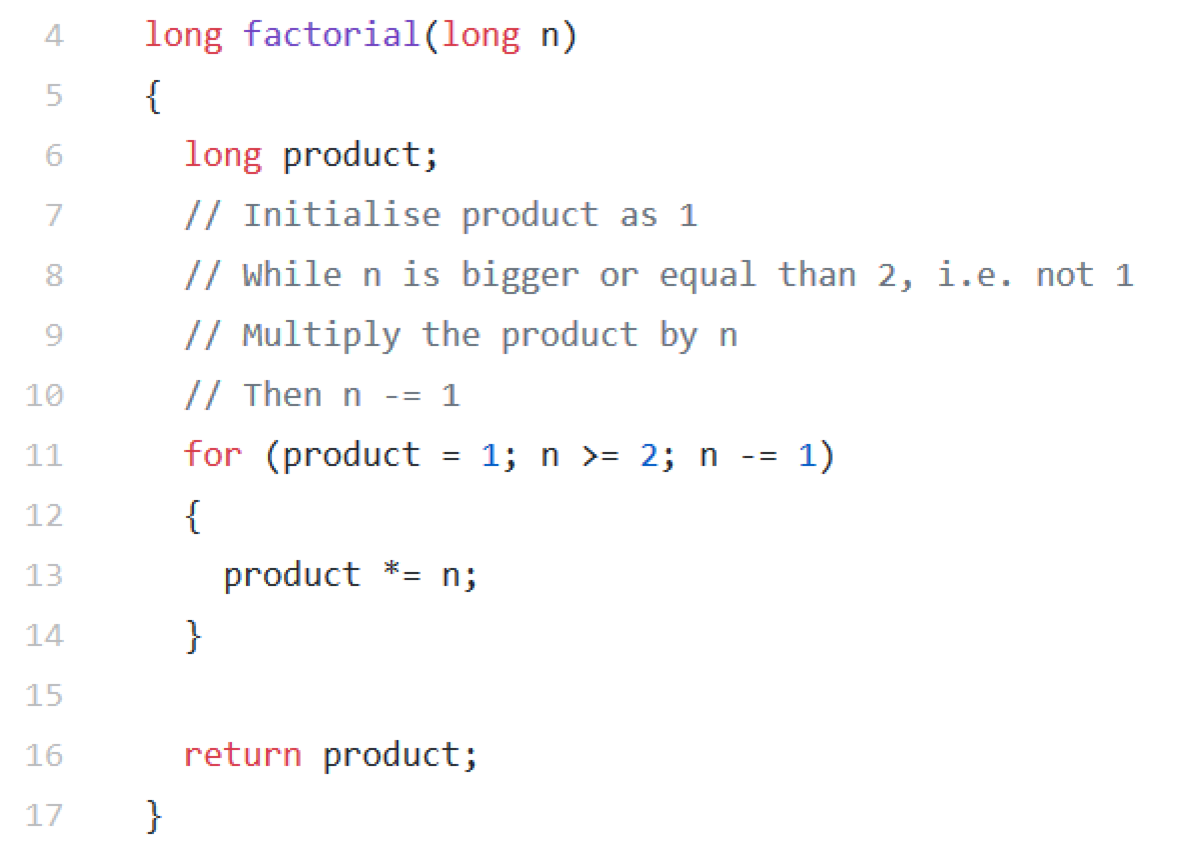
Does this code runs correctly? No



```c
long factorial(long n)
{
    long i = n - 1;
    long product;
    for (product = n; i >= 2; product *= i);
    {
        i -= 1;
    }
    return product;
}
```

Problem Set 11.1

Solution (Code available on GitHub repo)



```
long factorial(long n)
{
  long product;
  // Initialise product as 1
  // While n is bigger or equal than 2, i.e. not 1
  // Multiply the product by n
  // Then n -= 1
  for (product = 1; n >= 2; n -= 1)
  {
    product *= n;
  }

  return product;
}
```

PROBLEM SETS

10.1, 10.2, 11.1, 11.2, 11.3, 12.1

Problem Set 11.2

Code available on GitHub repo

a) Rewrite the "Guess A Number" program so that it shows the user the number of guesses made before the correct guess is entered.

b) Rewrite the "Guess A Number" program with a while loop.

c) Extend the "Guess A Number" program so that it plays the game for five rounds with the user, and at the end, shows the user the average number of guesses over five rounds. (Hint: you should put the loop that reads the guess and prints feedback to the user into another function.)

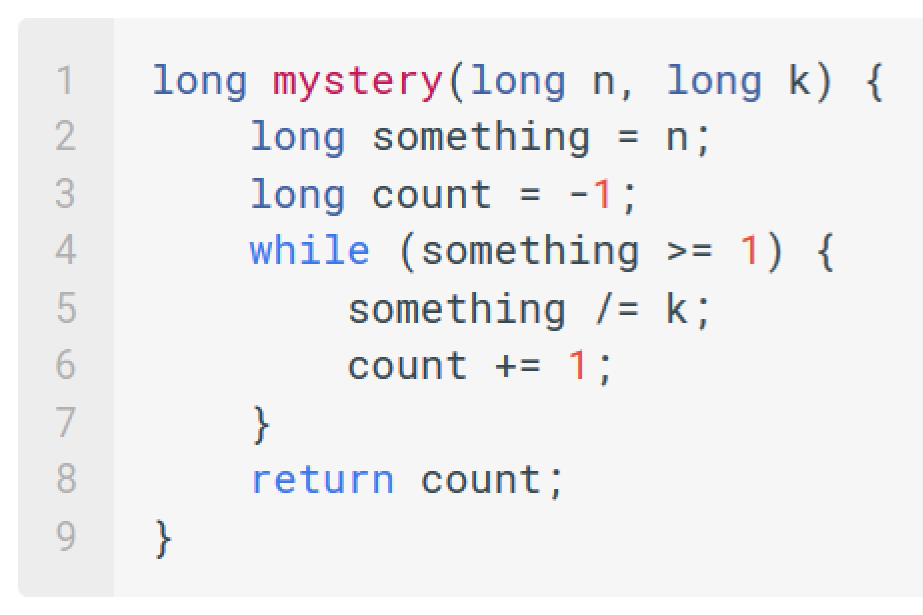
d) What is the optimal strategy to play the game?

PROBLEM SETS

10.1, 10.2, 11.1, 11.2, 11.3, 12.1

Problem Set 11.3 (a)

Trace the following algorithm



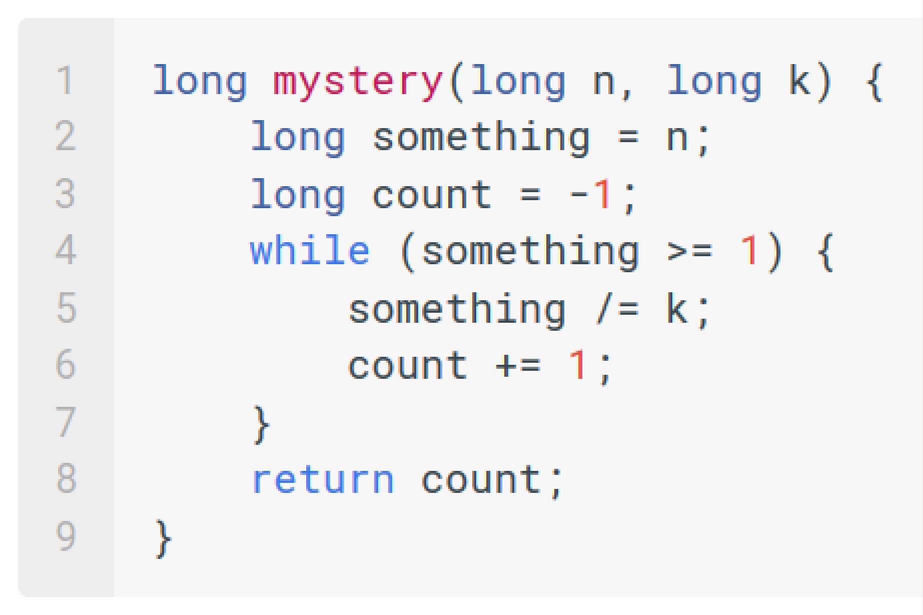
```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

What is the return value when:

- n is 8 and k is 2?
- n is 81 and k is 3?
- n is 100 and k is 5?

Problem Set 11.3 (a)

Trace the following algorithm



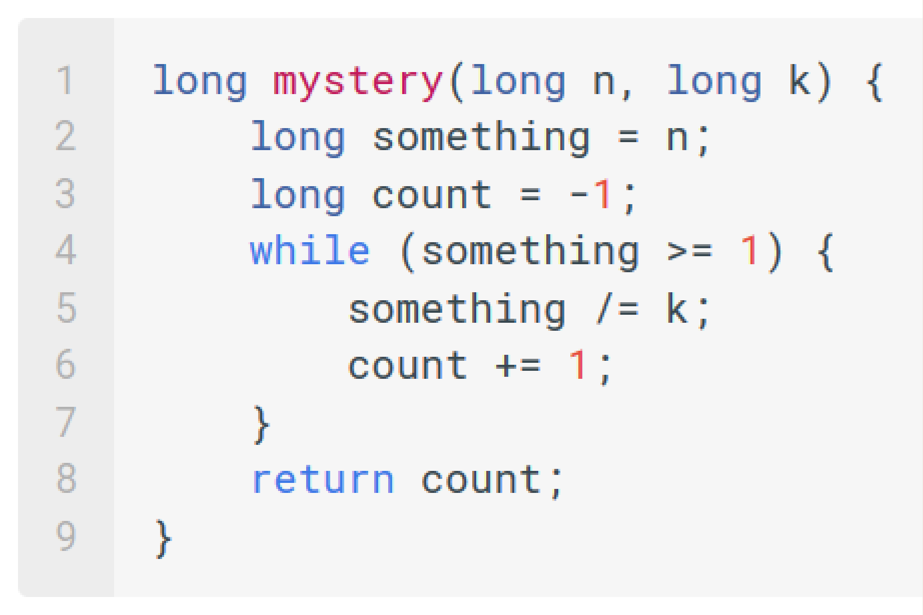
```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

What is the return value when:

- n is 8 and k is 2?
  - *3*
- n is 81 and k is 3?
  - *4*
- n is 100 and k is 5?
  - *2*

Problem Set 11.3 (b)

What is the mathematical expression that our mystery function here is trying to compute?

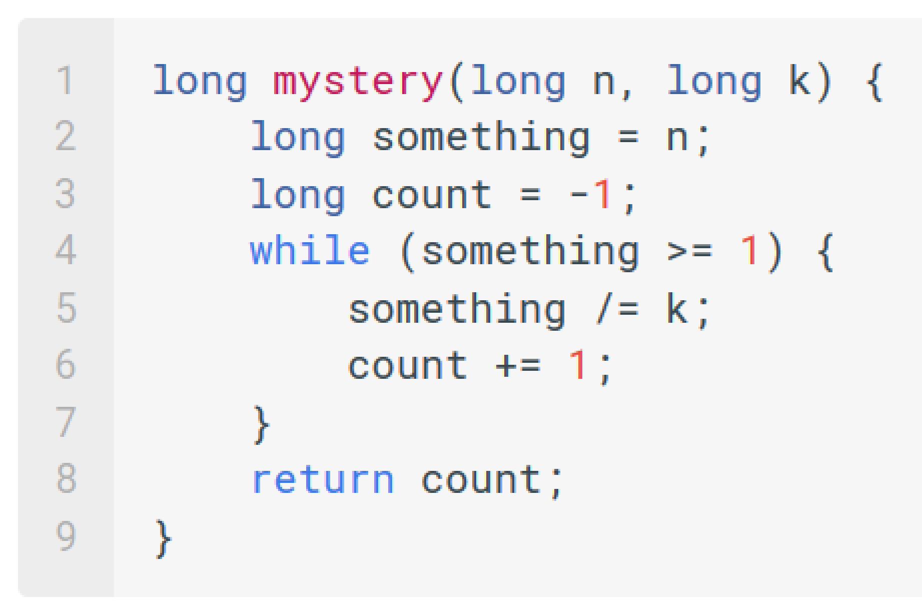


```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

It is: floor of log_k(n)

Problem Set 11.3 (c)

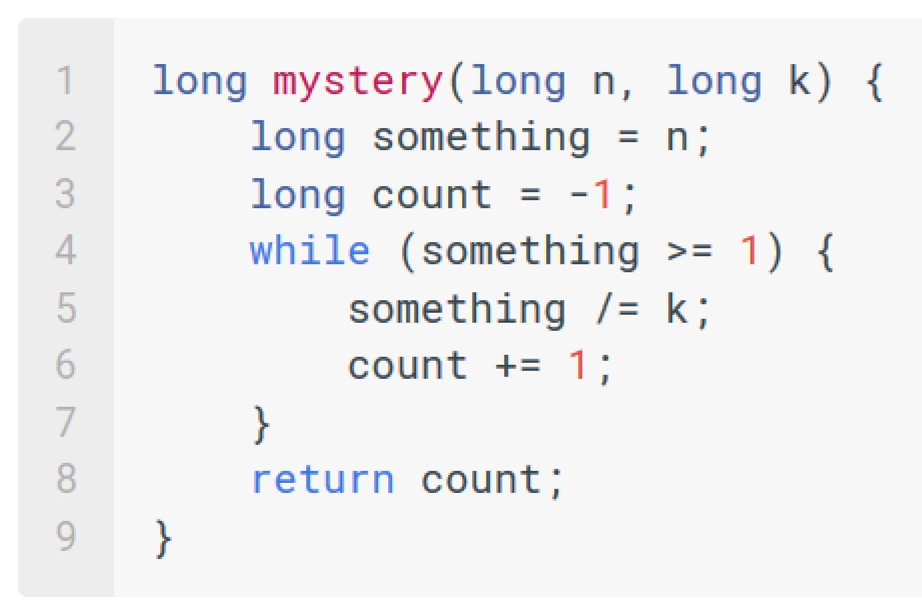
Give a pair of inputs that would cause the function to return the wrong answer.



```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

Problem Set 11.3 (c)

Give a pair of inputs that would cause the function to return the wrong answer.

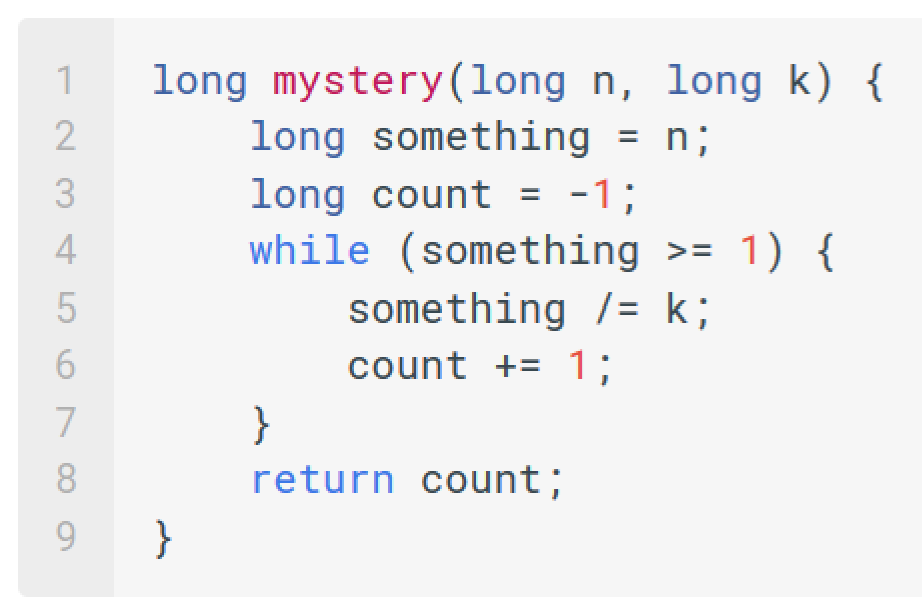


```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

If n is 0, then the value returned would be -1 (it should be undefined).

Problem Set 11.3 (d)

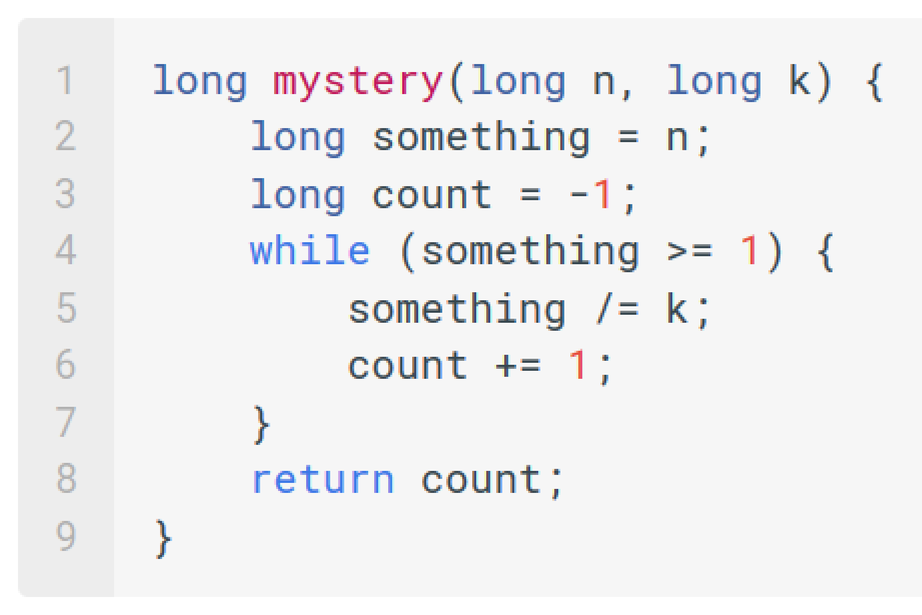
Give a pair of inputs that would cause the function to loop forever.



```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

Problem Set 11.3 (d)

Give a pair of inputs that would cause the function to loop forever.



```
long mystery(long n, long k) {
    long something = n;
    long count = -1;
    while (something >= 1) {
        something /= k;
        count += 1;
    }
    return count;
}
```

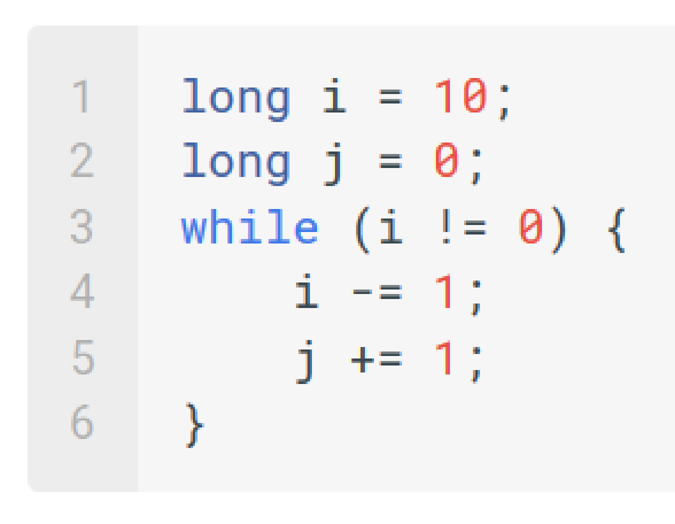
If k is 1, then something over k never changes, so it would loop forever (it should be undefined).

PROBLEM SETS

10.1, 10.2, 11.1, 11.2, 11.3, 12.1

Problem Set 12.1

Give a pair of inputs that would cause the function to loop forever.

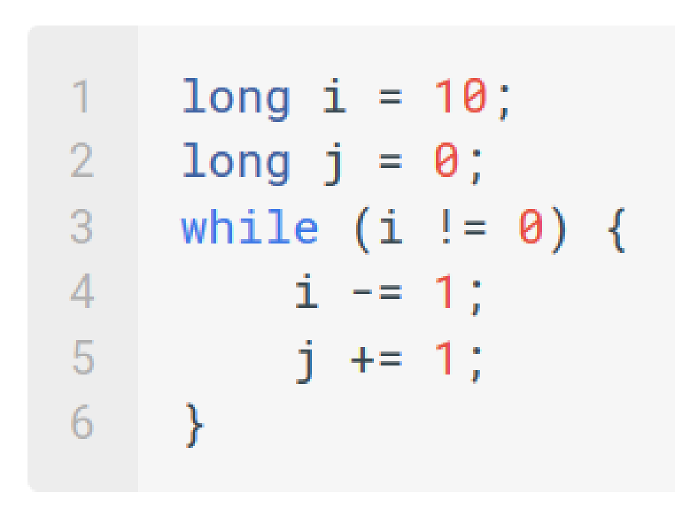


```
long i = 10;
long j = 0;
while (i != 0) {
    i -= 1;
    j += 1;
}
```

- (a) Trace through the program. What is the value of j when the loop exits?
- (b) Do you recognize any pattern on the relationship of i and j?
- (C) What is the loop invariant?

Problem Set 12.1

Give a pair of inputs that would cause the function to loop forever.



```
long i = 10;
long j = 0;
while (i != 0) {
    i -= 1;
    j += 1;
}
```

- (a) Trace through the program. What is the value of j when the loop exits?
  - *10*
- (b) Do you recognize any pattern on the relationship of i and j?
  - *{ i + j == 10 }*
- (c) What is the loop invariant?
  - *{ i + j == 10 }*

THE END

https://github.com/DigiPie/cs1010_tut_c09