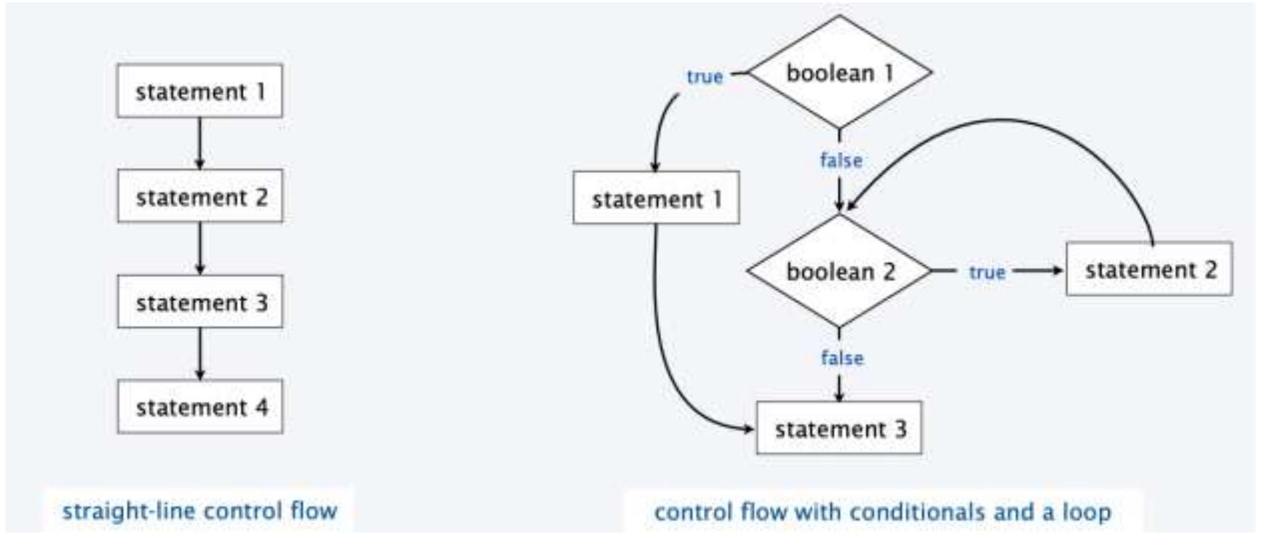
Conditionals and Loops

Ruilizhen HU

122090168@link.cuhk.edu.cn

Overview

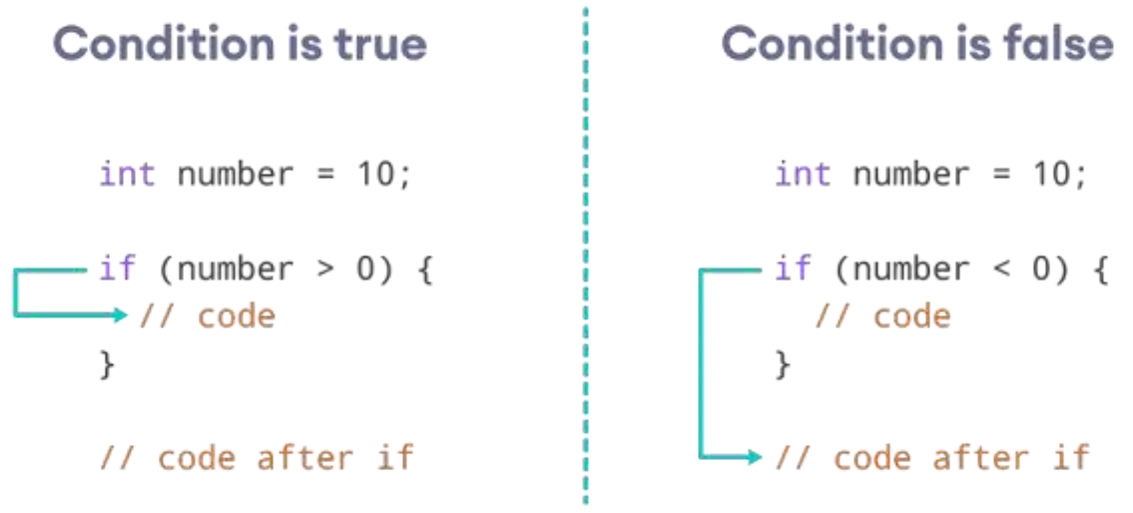
- Conditionals
  - The if statement
- Loops
  - The while statement
  - The for loop
  - The do while statement
- Nesting
- Debugging



The if statement

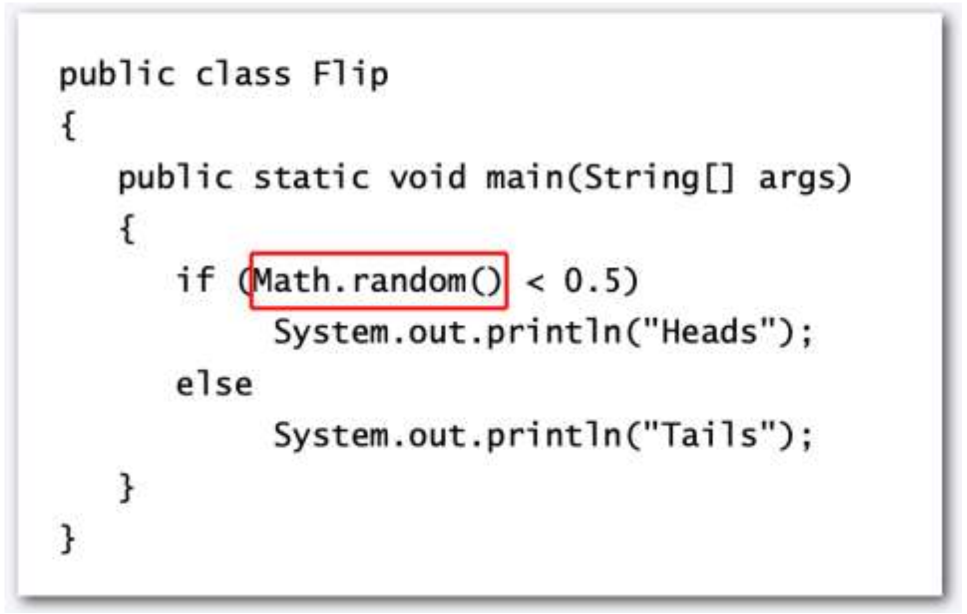
Execute certain statements depending on the values of certain variables

- Evaluate a boolean expression
- If true, execute a statement
- The else option: If false, execute a different statement



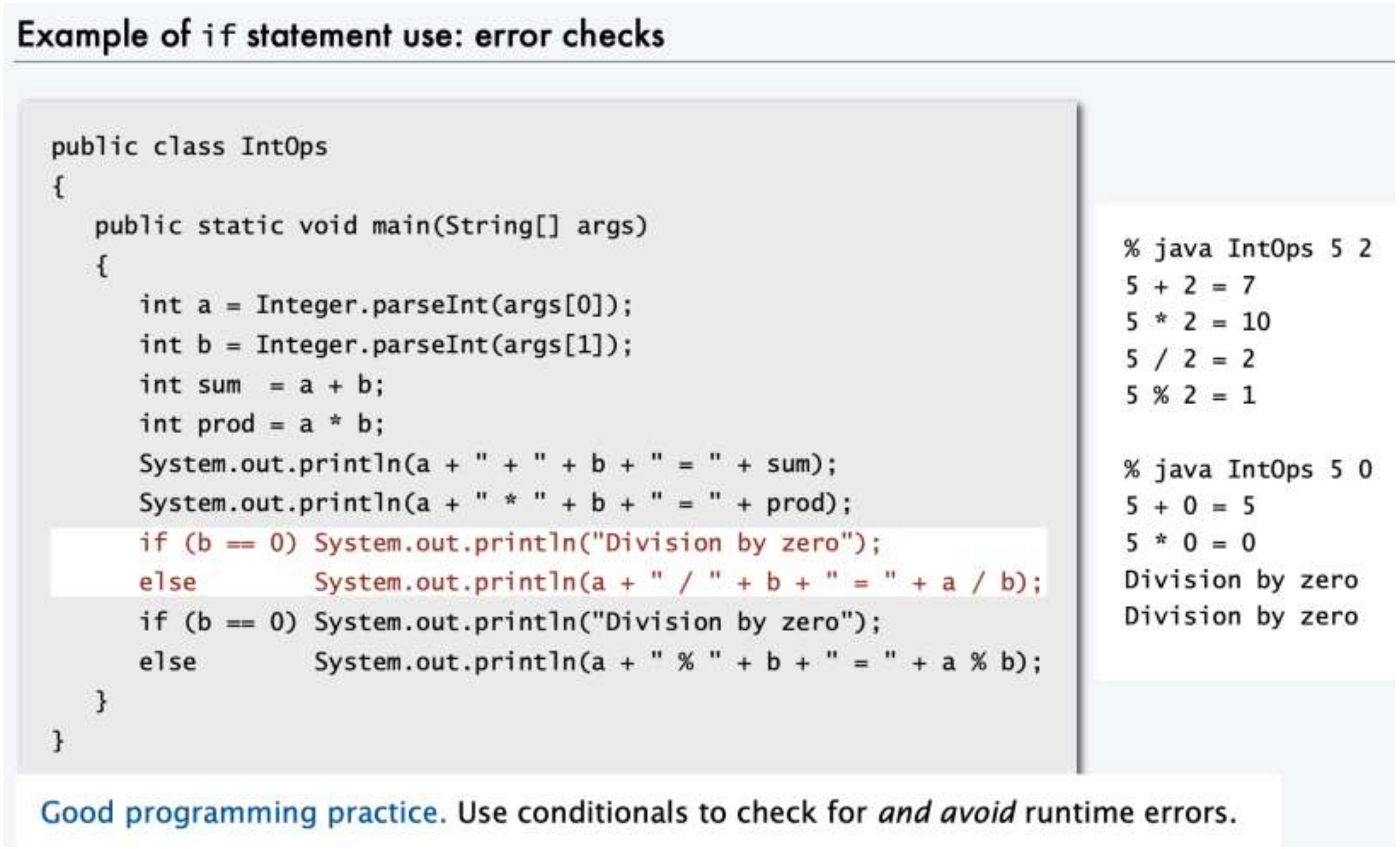
```
int number = 10;

if (number > 0) {
  // code
}

// code after if
```

```
int number = 10;

if (number < 0) {
  // code
}

// code after if
```



```
public class Flip
{
   public static void main(String[] args)
   {
      if (Math.random() < 0.5)
           System.out.println("Heads");
      else
           System.out.println("Tails");
   }
}
```

The if statement



Example of if statement use: error checks

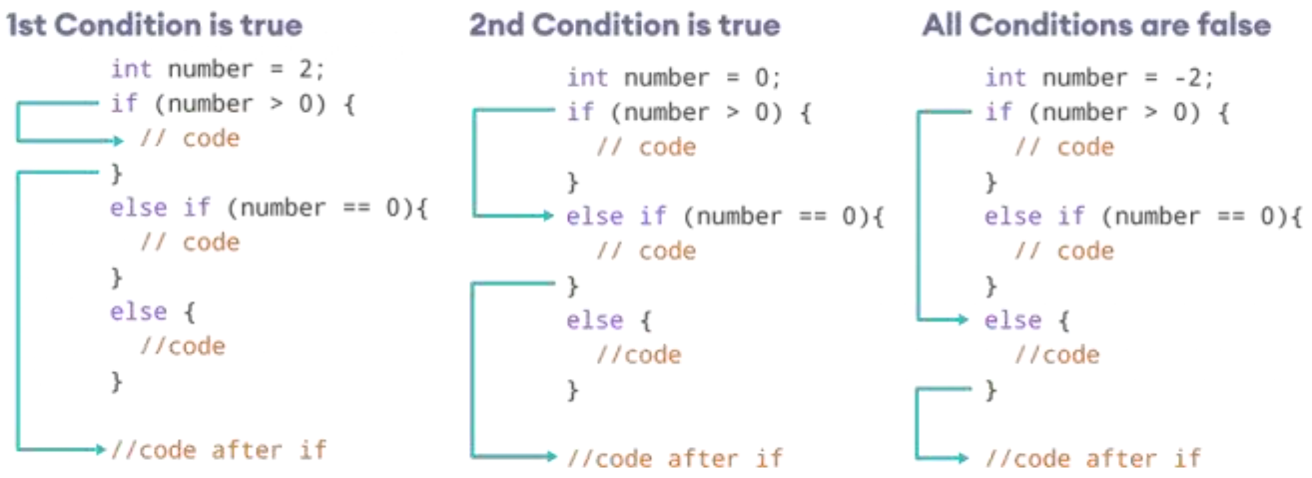
```
public class IntOps
{
   public static void main(String[] args)
   {
      int a = Integer.parseInt(args[0]);
      int b = Integer.parseInt(args[1]);
      int sum  = a + b;
      int prod = a * b;
      System.out.println(a + " + " + b + " = " + sum);
      System.out.println(a + " * " + b + " = " + prod);
      if (b == 0) System.out.println("Division by zero");
      else        System.out.println(a + " / " + b + " = " + a / b);
      if (b == 0) System.out.println("Division by zero");
      else        System.out.println(a + " % " + b + " = " + a % b);
   }
}
```

```
% java IntOps 5 2
5 + 2 = 7
5 * 2 = 10
5 / 2 = 2
5 % 2 = 1

% java IntOps 5 0
5 + 0 = 5
5 * 0 = 0
Division by zero
Division by zero
```

Good programming practice. Use conditionals to check for *and avoid* runtime errors.

The if statement - Extension

else if



1st Condition is true

```
int number = 2;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

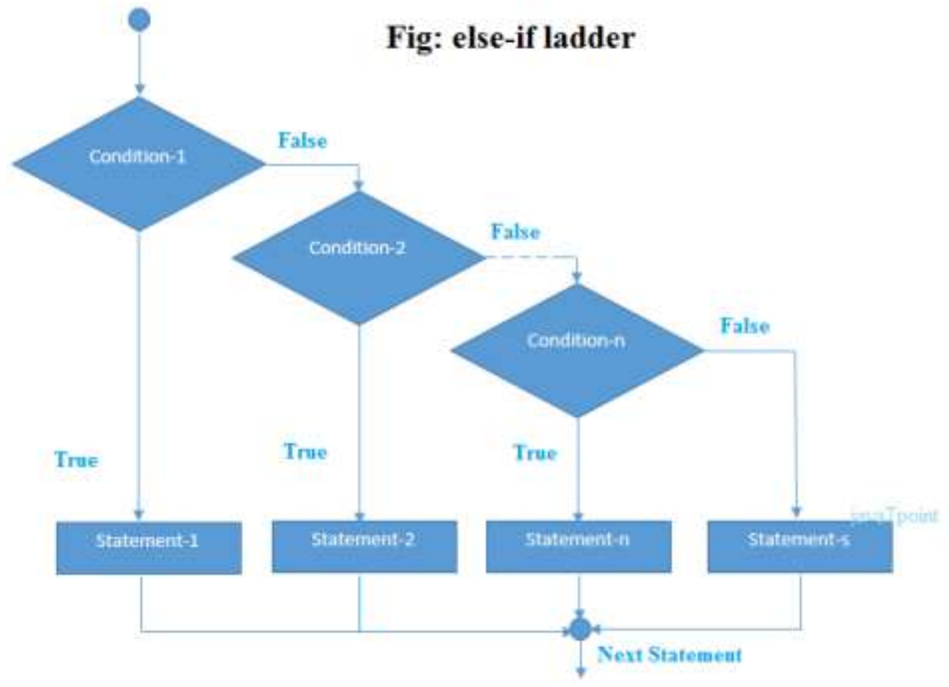
2nd Condition is true

```
int number = 0;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

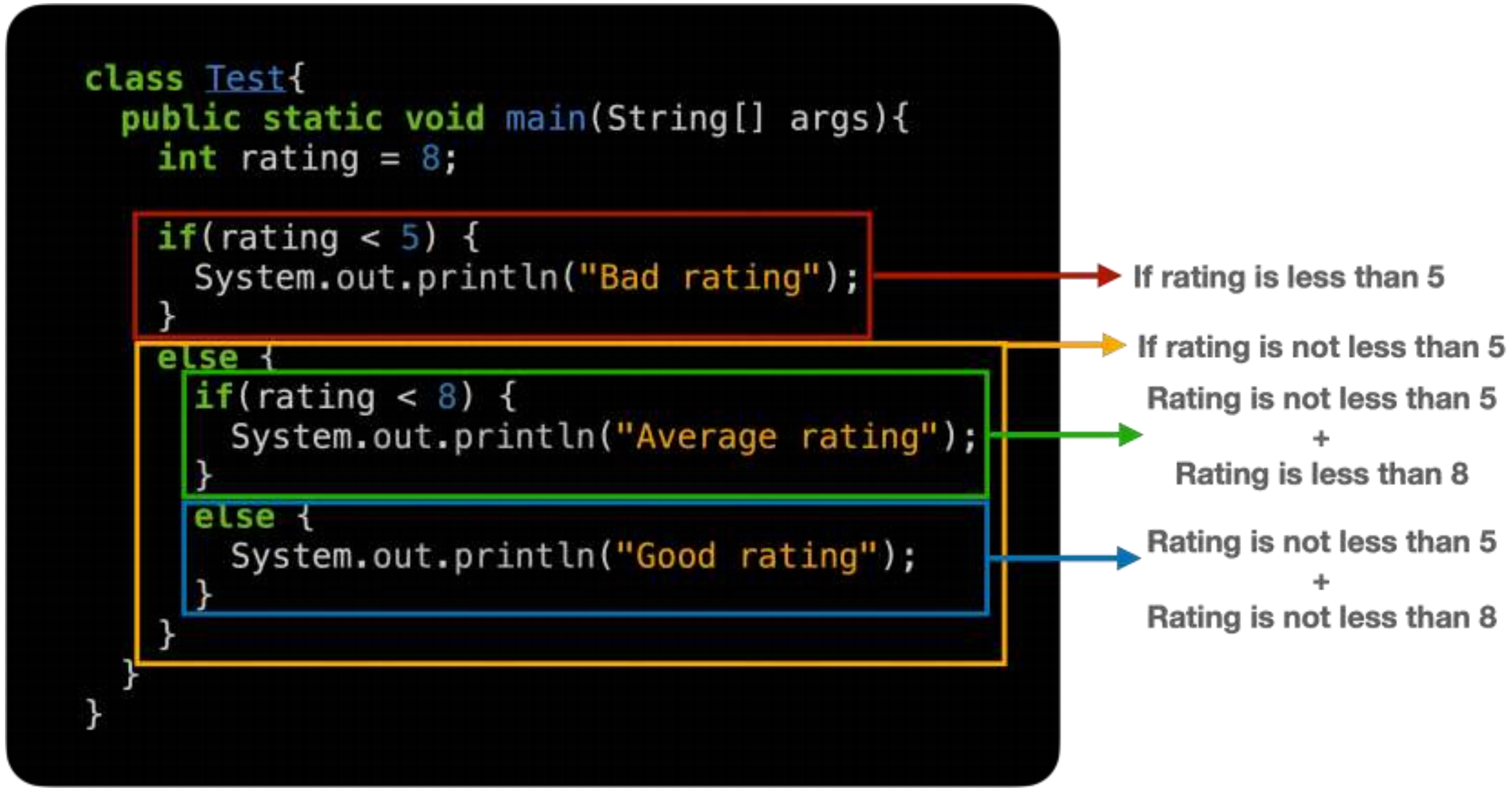
All Conditions are false

```
int number = -2;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

Fig: else-if ladder



The if statement - Extension



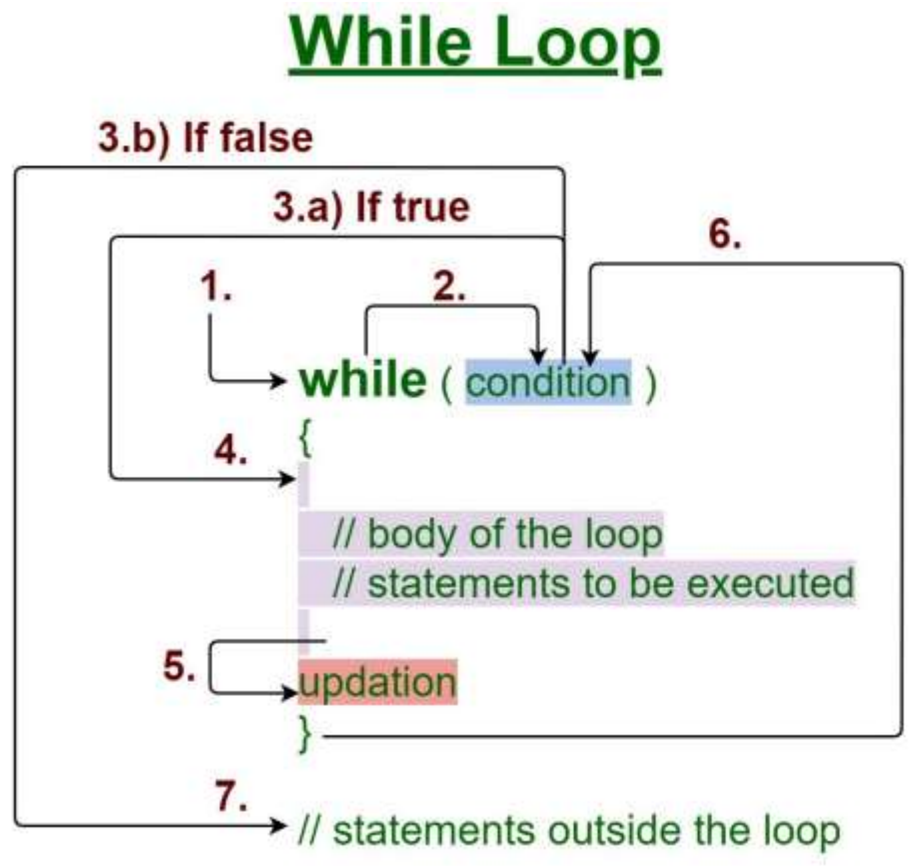
```
class Test{
  public static void main(String[] args){
    int rating = 8;

    if(rating < 5) {
      System.out.println("Bad rating");
    }
    else {
      if(rating < 8) {
        System.out.println("Average rating");
      }
      else {
        System.out.println("Good rating");
      }
    }
  }
}
```

If rating is less than 5
If rating is not less than 5
Rating is not less than 5
+
Rating is less than 8
Rating is not less than 5
+
Rating is not less than 8

The while statement

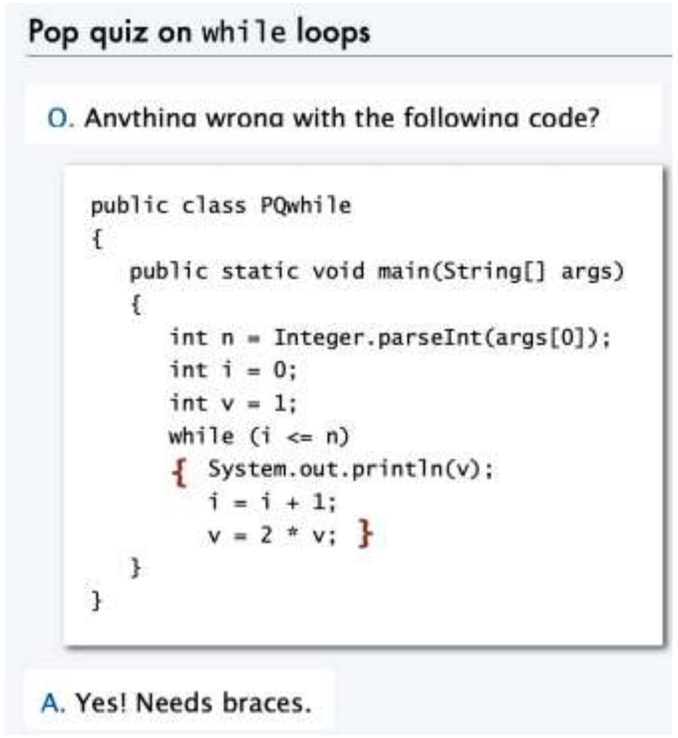
Execute certain statements repeatedly until certain conditions are met

- Evaluate a boolean expression
- If true, execute a sequence of statements
- Repeat



The while statement - Cautions

- Add braces
- Do not add the semicolon after the “while”

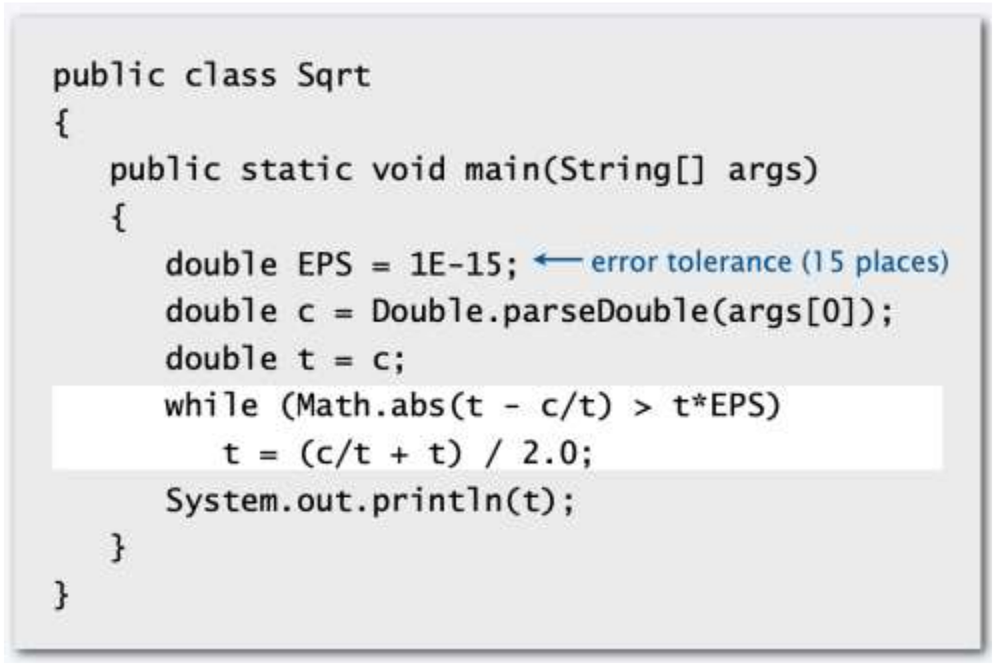


Pop quiz on while loops

Q. Anything wrong with the following code?

```
public class PQwhile
{
   public static void main(String[] args)
   {
      int n = Integer.parseInt(args[0]);
      int i = 0;
      int v = 1;
      while (i <= n)
      { System.out.println(v);
         i = i + 1;
         v = 2 * v; }
   }
}
```

A. Yes! Needs braces.



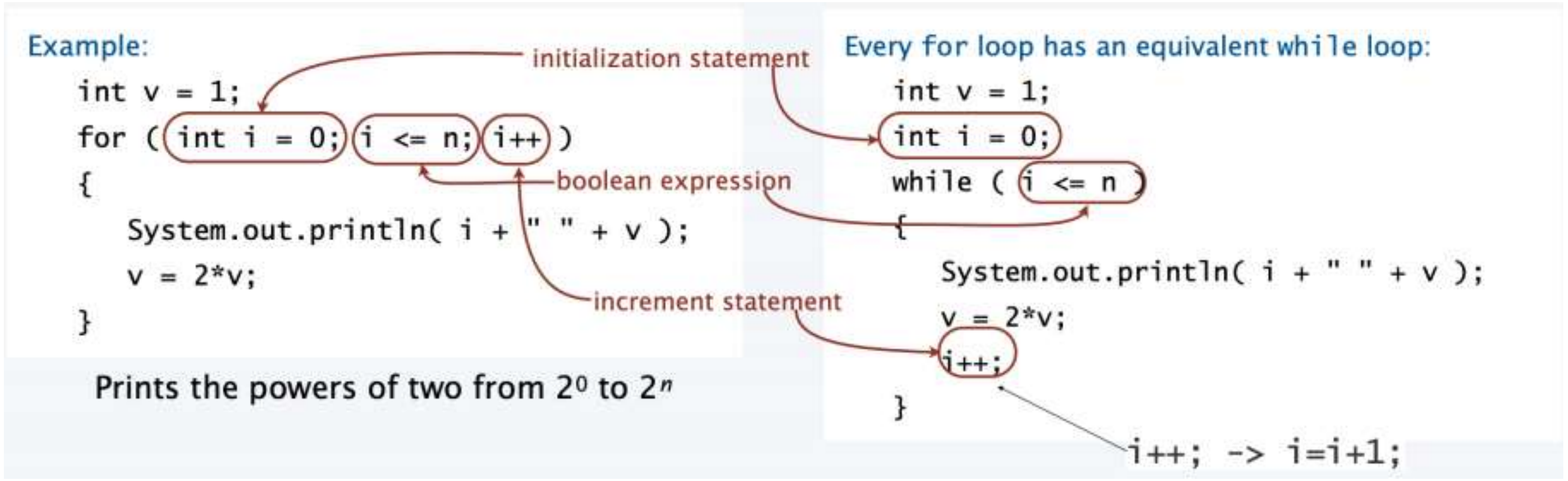
```
public class Sqrt
{
   public static void main(String[] args)
   {
      double EPS = 1E-15;  ← error tolerance (15 places)
      double c = Double.parseDouble(args[0]);
      double t = c;
      while (Math.abs(t - c/t) > t*EPS)
         t = (c/t + t) / 2.0;
      System.out.println(t);
   }
}
```

```
int i = 0;
while ( i < 999);
{
   System.out.println("hello");
    i = i + 1;
}
```

The for loop

An alternative repetition structure

- Evaluate an initialization statement
- Evaluate a boolean expression
- If true, execute a sequence of statements, then execute an increment statement
- Repeat



Example:

```
int v = 1;
for (int i = 0; i <= n; i++)
{
   System.out.println( i + " " + v );
   v = 2*v;
}
```

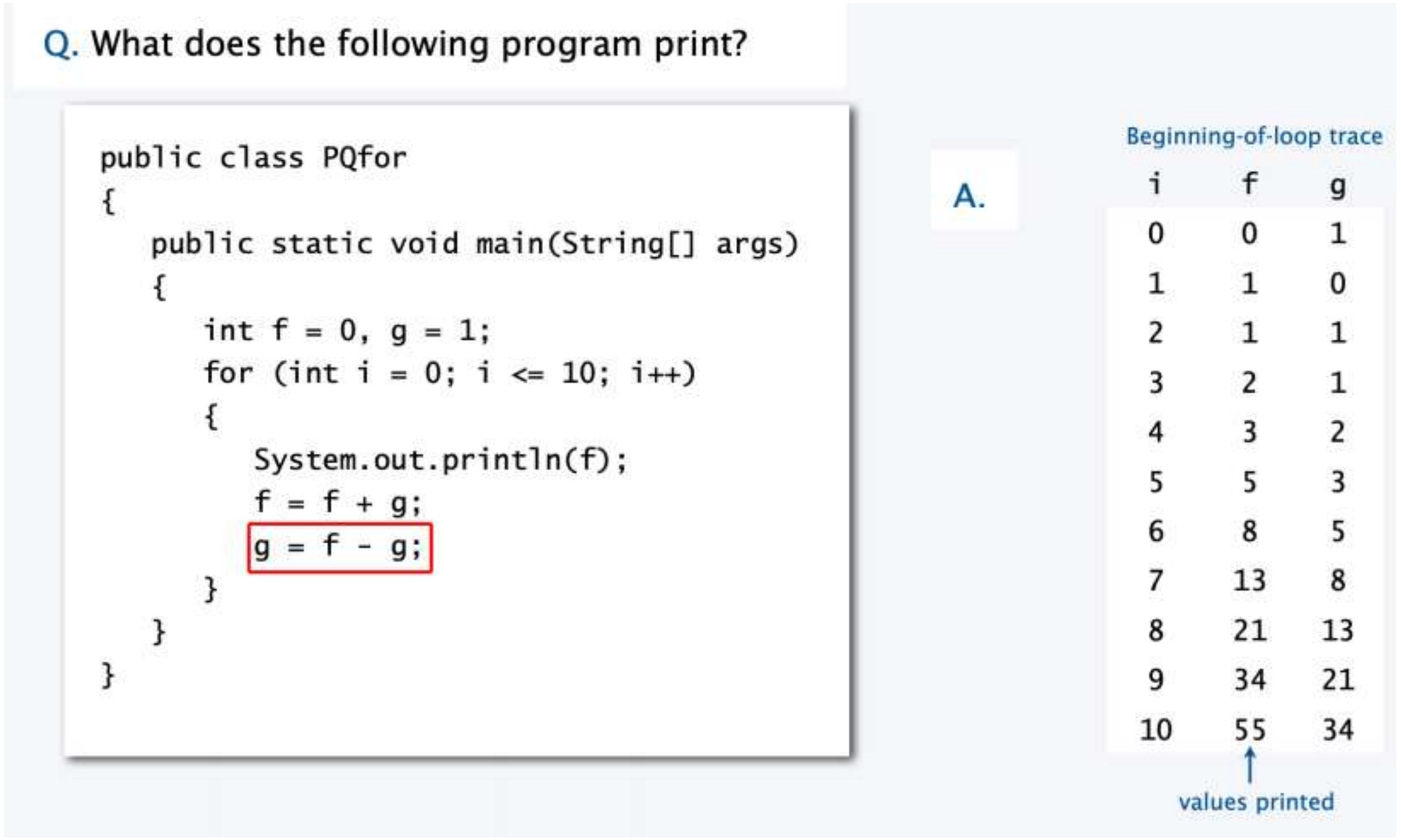
Prints the powers of two from 2^0 to 2^n

Every for loop has an equivalent while loop:

```
int v = 1;
int i = 0;
while ( i <= n )
{
   System.out.println( i + " " + v );
   v = 2*v;
   i++;
}
```

```
i++; -> i=i+1;
```

The for loop



Q. What does the following program print?

```
public class PQfor
{
   public static void main(String[] args)
   {
      int f = 0, g = 1;
      for (int i = 0; i <= 10; i++)
      {
         System.out.println(f);
         f = f + g;
         g = f - g;
      }
   }
}
```

A.

Beginning-of-loop trace

i	f	g
0	0	1
1	1	0
2	1	1
3	2	1
4	3	2
5	5	3
6	8	5
7	13	8
8	21	13
9	34	21
10	55	34

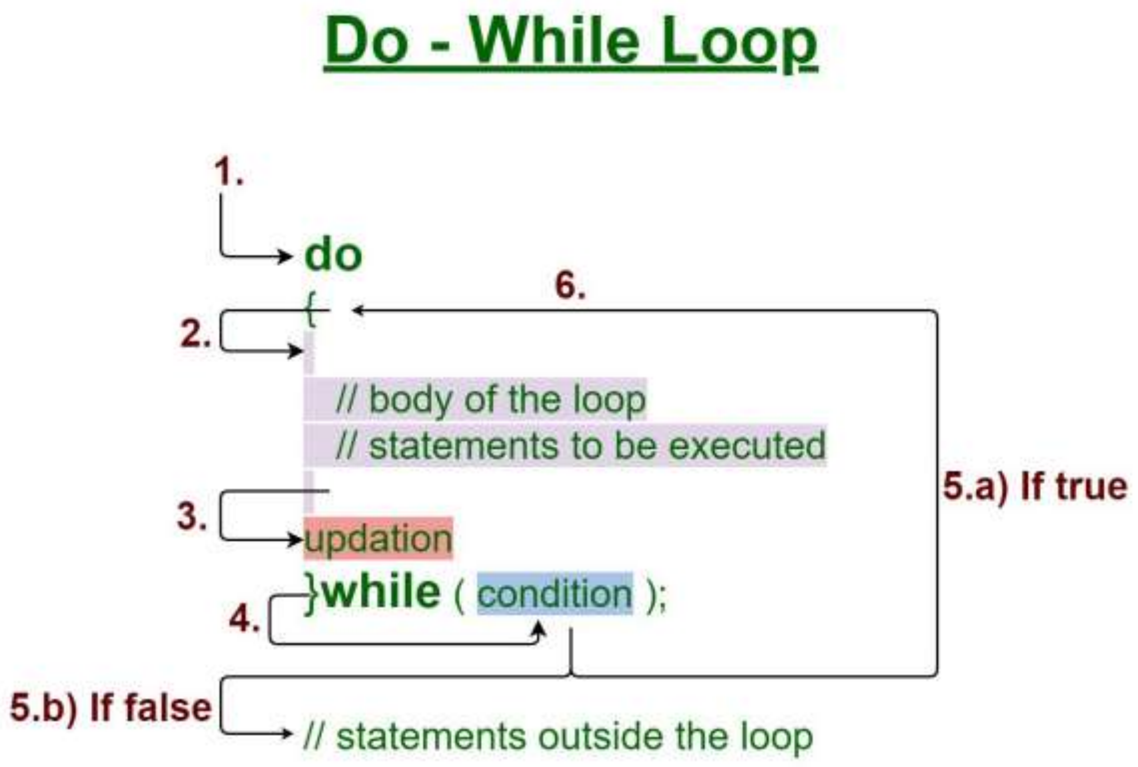
↑ values printed

Fibonacci Sequence

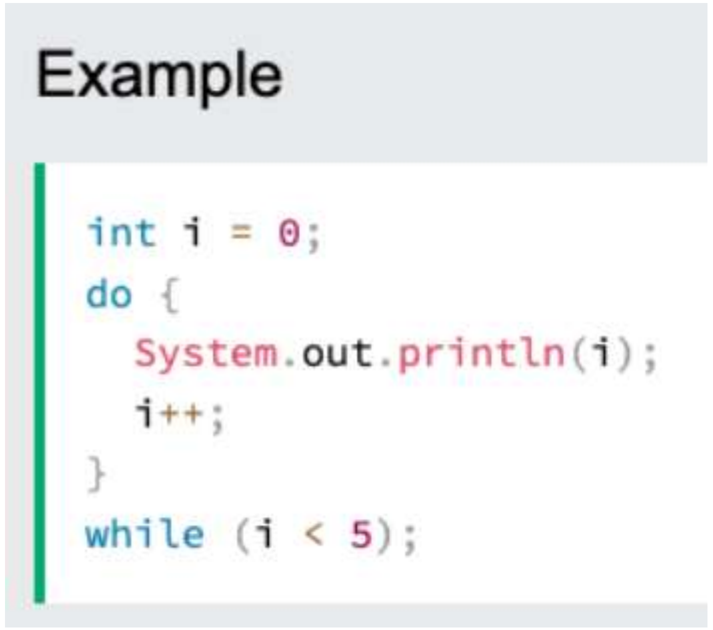
The do while statement

The do/while loop is a variant of the while loop

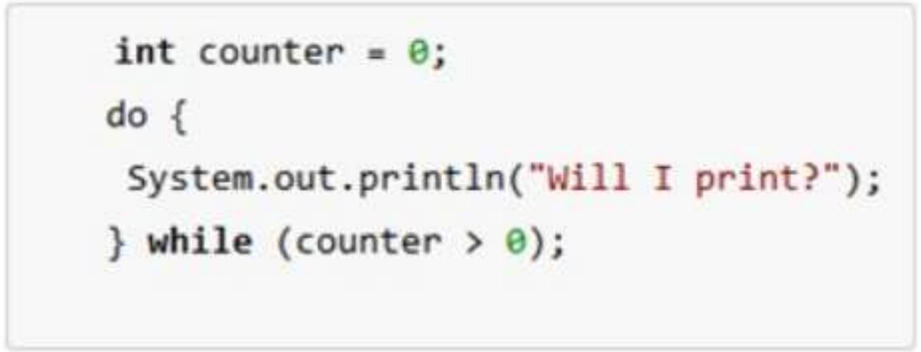
- Execute the code block once before checking if the condition is true
- Evaluate a boolean expression
- If true, execute the code block again
- Repeat



The do while statement


Example

```
int i = 0;
do {
  System.out.println(i);
  i++;
}
while (i < 5);
```

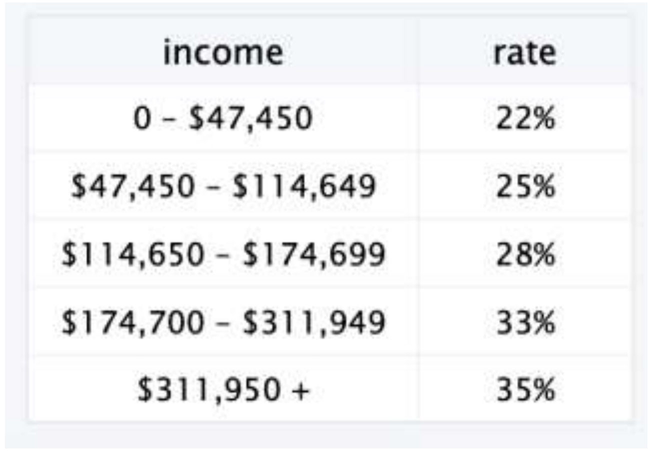


```
 int counter = 0;
do {
 System.out.println("Will I print?");
} while (counter > 0);
```

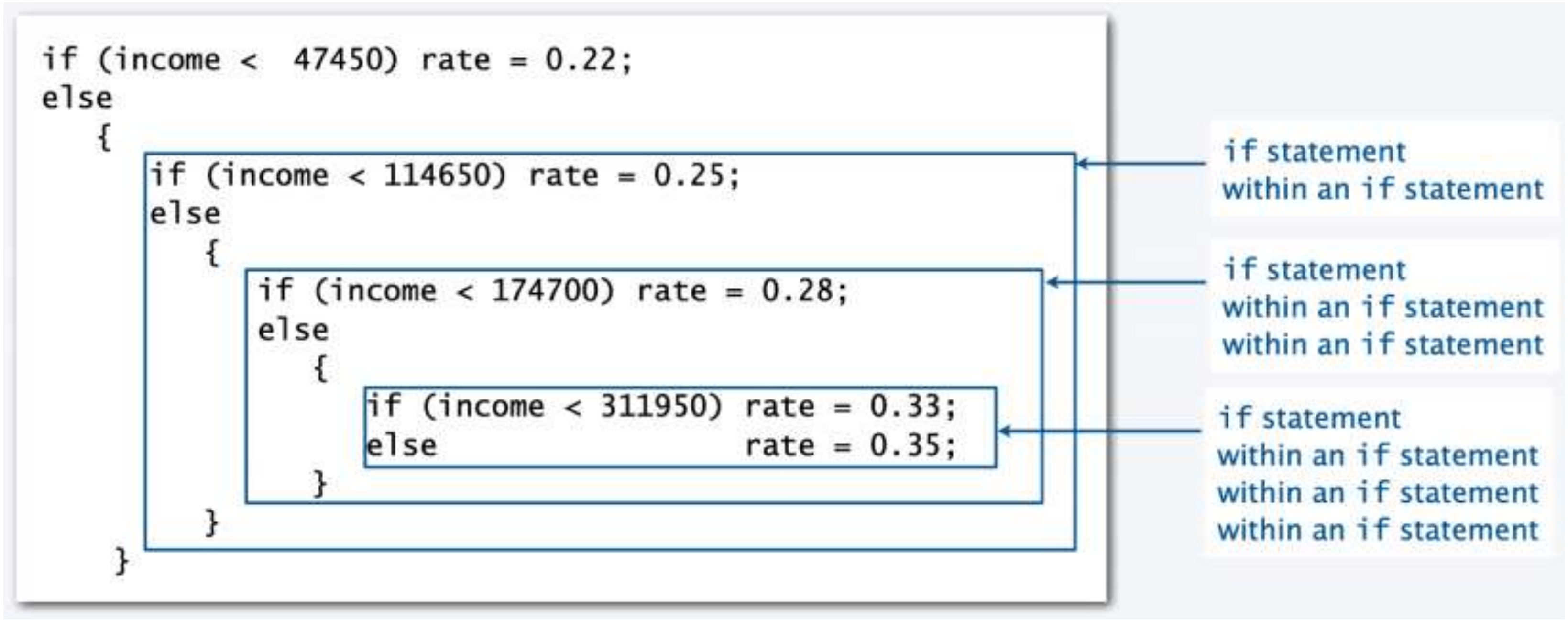
Note: The do while statements can always be rewritten as the while statement

Nesting

- Any "statement" within a conditional or loop may itself be a conditional or a loop statement
- Enables complex control flows
- Adds to challenge of debugging



income	rate
0 – $47,450	22%
$47,450 – $114,649	25%
$114,650 – $174,699	28%
$174,700 – $311,949	33%
$311,950 +	35%

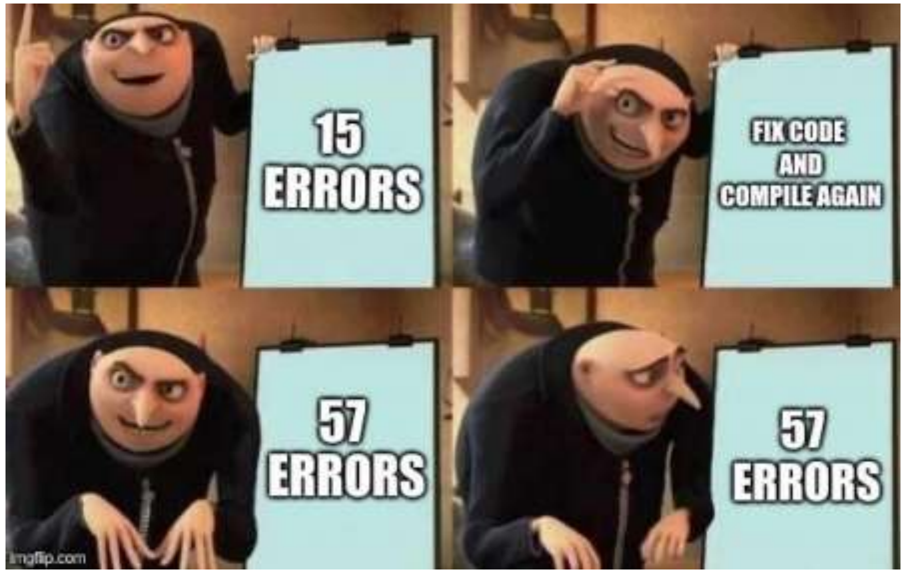


```
if (income <  47450) rate = 0.22;
else
   {
     if (income < 114650) rate = 0.25;
     else
        {
          if (income < 174700) rate = 0.28;
          else
             {
               if (income < 311950) rate = 0.33;
               else                 rate = 0.35;
             }
        }
   }
```

if statement within an if statement
if statement within an if statement within an if statement
if statement within an if statement within an if statement within an if statement

Debugging

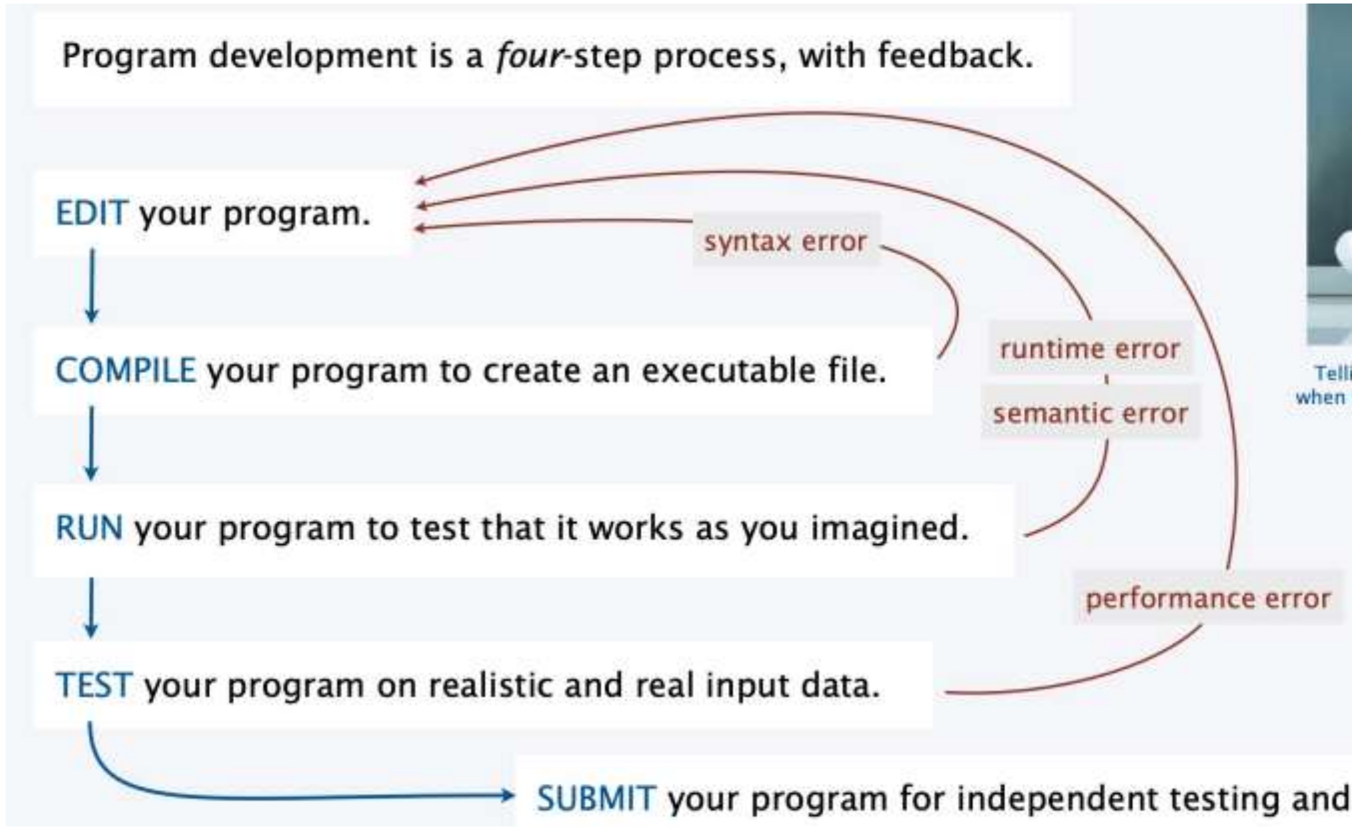
Programmers. Everyday.
Why?
Why?
Why?
Oh, that's why.


15 ERRORS
FIX CODE AND COMPILE AGAIN
57 ERRORS
57 ERRORS
imgflip.com

ITS DONE....
DEBUGGING IS DONE

Debugging

- Bug: A mistake in a program
- Debugging: The process of eliminating bugs



Program development is a *four*-step process, with feedback.

EDIT your program.

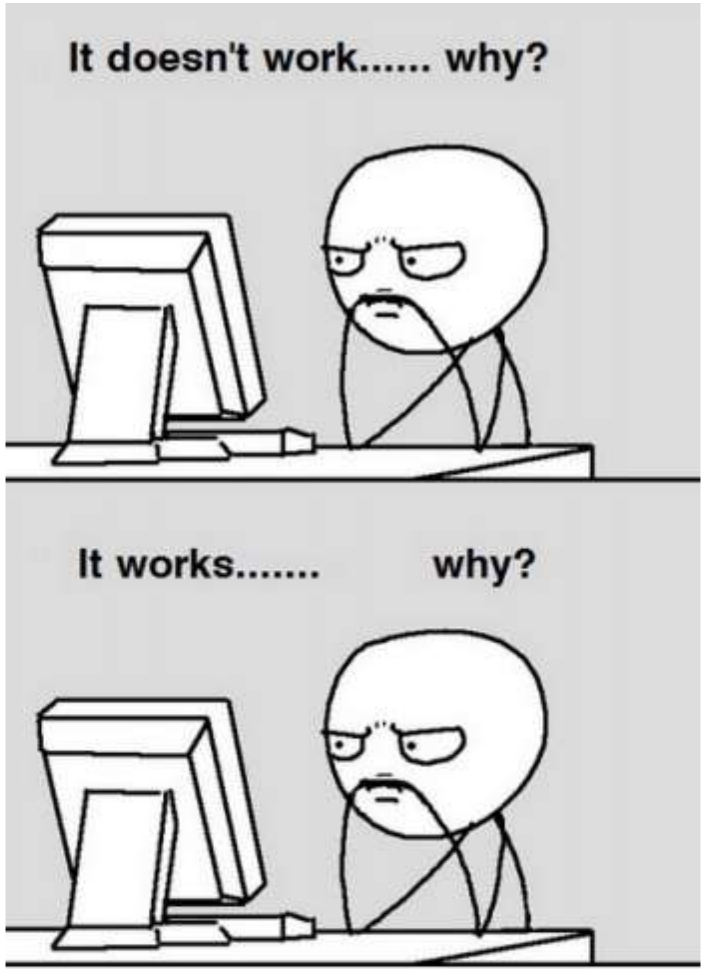
COMPILE your program to create an executable file.

RUN your program to test that it works as you imagined.

TEST your program on realistic and real input data.

SUBMIT your program for independent testing and approval.

Telling a computer what to do when you know what you're doing

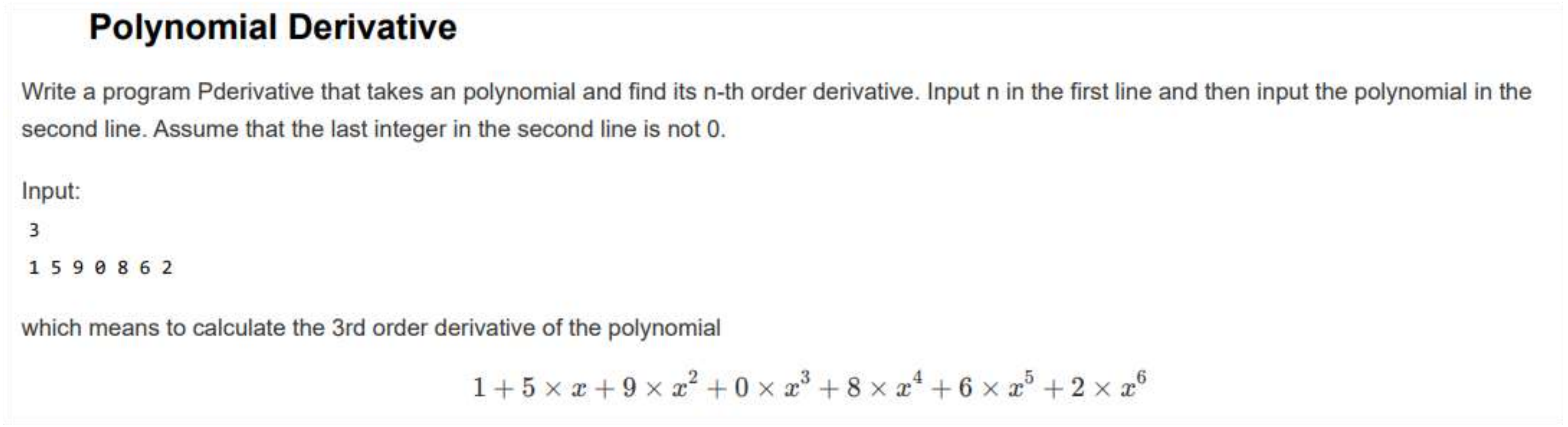


Debugging

Debugging the Eclipse IDE for Java Developers

Tutorial: Debug your first Java application

Problem 1



Polynomial Derivative

Write a program Pderivative that takes an polynomial and find its n-th order derivative. Input n in the first line and then input the polynomial in the second line. Assume that the last integer in the second line is not 0.

Input:

```
3
1 5 9 0 8 6 2
```

which means to calculate the 3rd order derivative of the polynomial

$$1 + 5 \times x + 9 \times x^2 + 0 \times x^3 + 8 \times x^4 + 6 \times x^5 + 2 \times x^6$$

OJ link: http://10.26.200.14/d/csc2003_2024_Spring/p/P10006

Problem 1

1. How do you differentiate it once

2. How do you apply the method multiple times

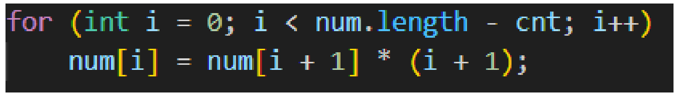
Problem 1

1. How do you differentiate it once

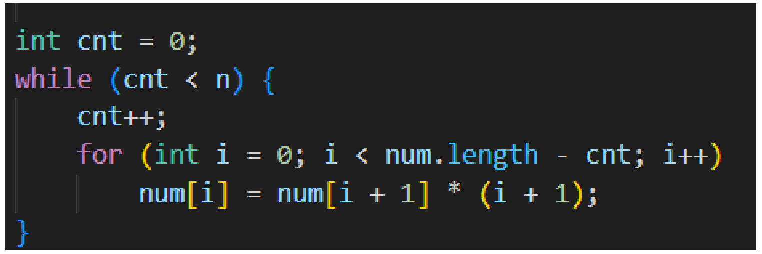
   for loop here

2. How do you apply the method multiple times

   while loop here

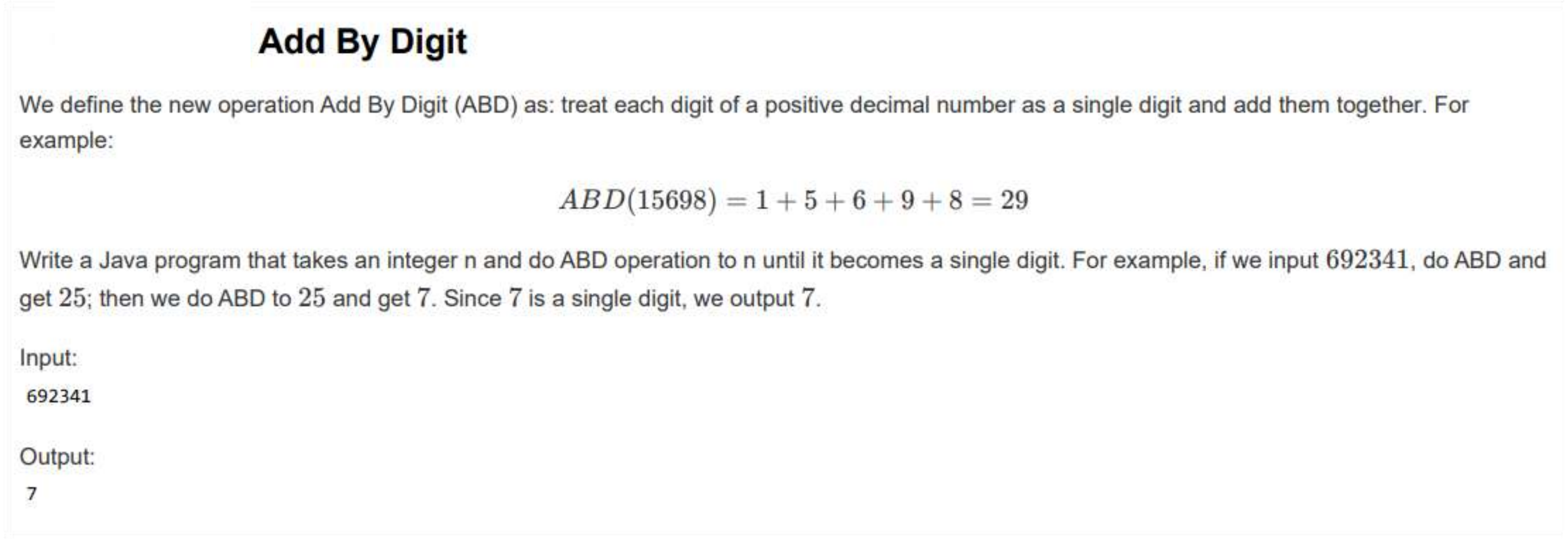


```
for (int i = 0; i < num.length - cnt; i++)
    num[i] = num[i + 1] * (i + 1);
```



```
int cnt = 0;
while (cnt < n) {
    cnt++;
    for (int i = 0; i < num.length - cnt; i++)
        num[i] = num[i + 1] * (i + 1);
}
```

Problem 2


Add By Digit

We define the new operation Add By Digit (ABD) as: treat each digit of a positive decimal number as a single digit and add them together. For example:

$$ABD(15698) = 1 + 5 + 6 + 9 + 8 = 29$$

Write a Java program that takes an integer n and do ABD operation to n until it becomes a single digit. For example, if we input 692341, do ABD and get 25; then we do ABD to 25 and get 7. Since 7 is a single digit, we output 7.

Input:

```
692341
```

Output:

```
7
```

OJ link: http://10.26.200.14/d/csc2003_2024_Spring/p/P10007

Problem 2

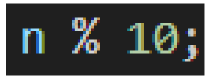
1. How to get a single digit from the given integer

2. How to apply the method multiple times

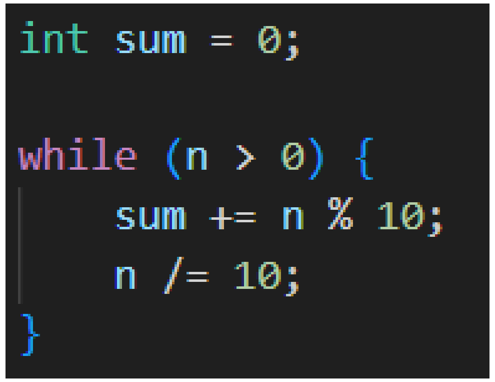
Problem 2

1. How to get a single digit from the given integer

2. How to apply the method multiple times

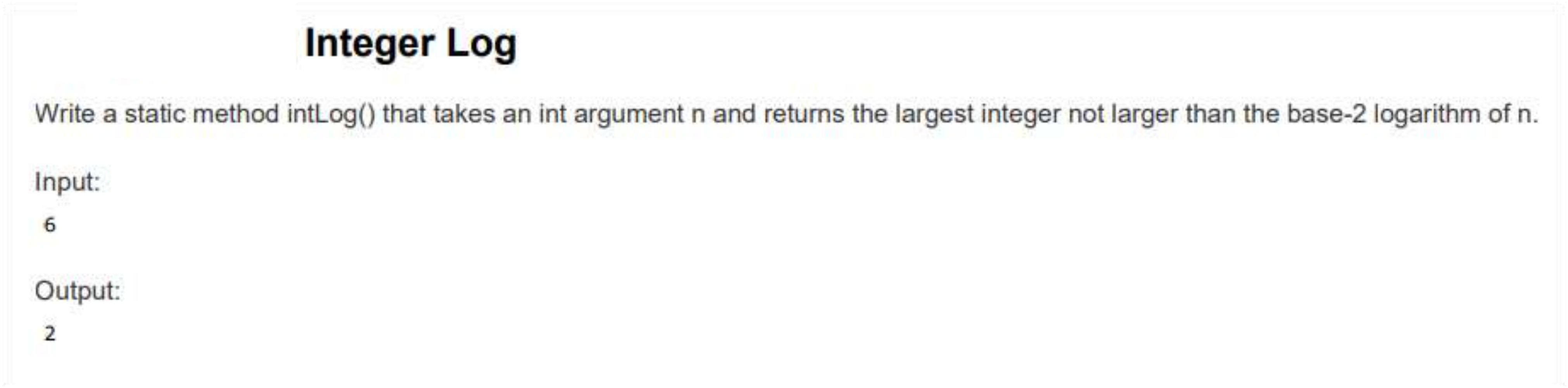


```
n % 10;
```



```
int sum = 0;

while (n > 0) {
    sum += n % 10;
    n /= 10;
}
```

Problem 3



Integer Log

Write a static method intLog() that takes an int argument n and returns the largest integer not larger than the base-2 logarithm of n.

Input:

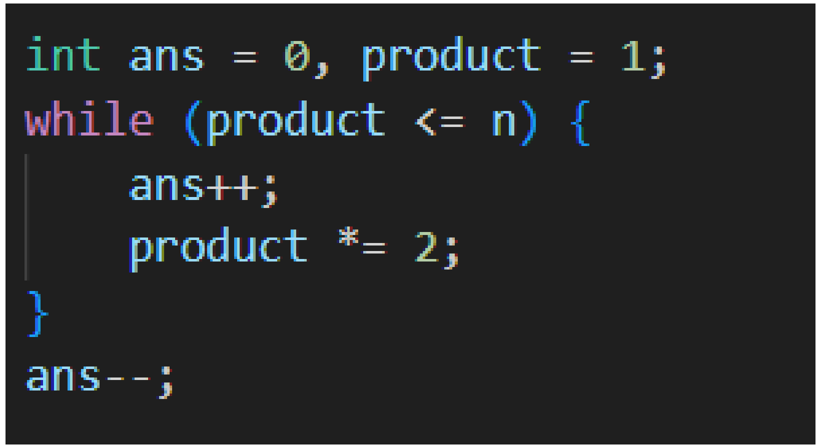
```
6
```

Output:

```
2
```

Problem 3

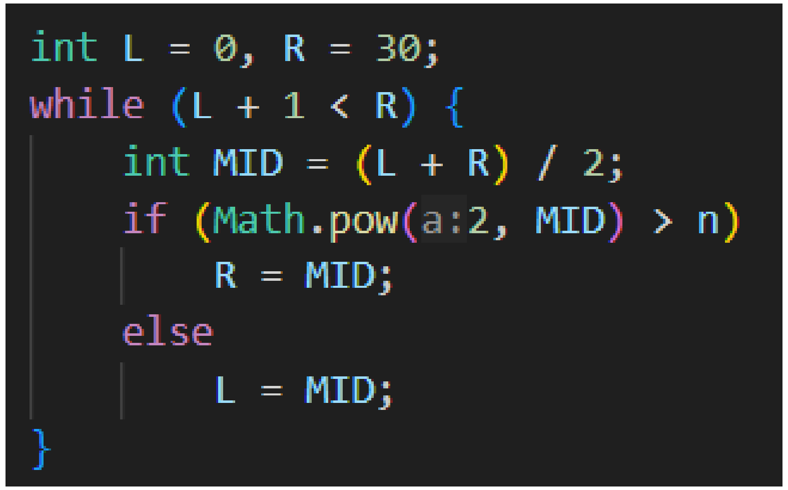
Enumerate one by one:



```
int ans = 0, product = 1;
while (product <= n) {
    ans++;
    product *= 2;
}
ans--;
```

Problem 3

Method of bisection:



```
int L = 0, R = 30;
while (L + 1 < R) {
    int MID = (L + R) / 2;
    if (Math.pow(a:2, MID) > n)
        R = MID;
    else
        L = MID;
}
```

Q&A