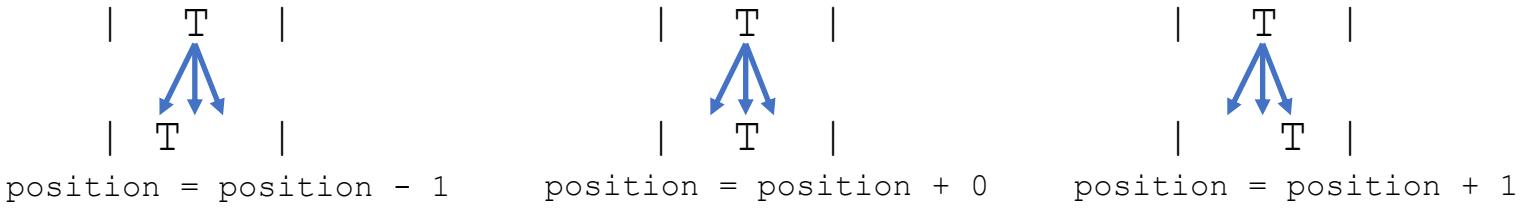
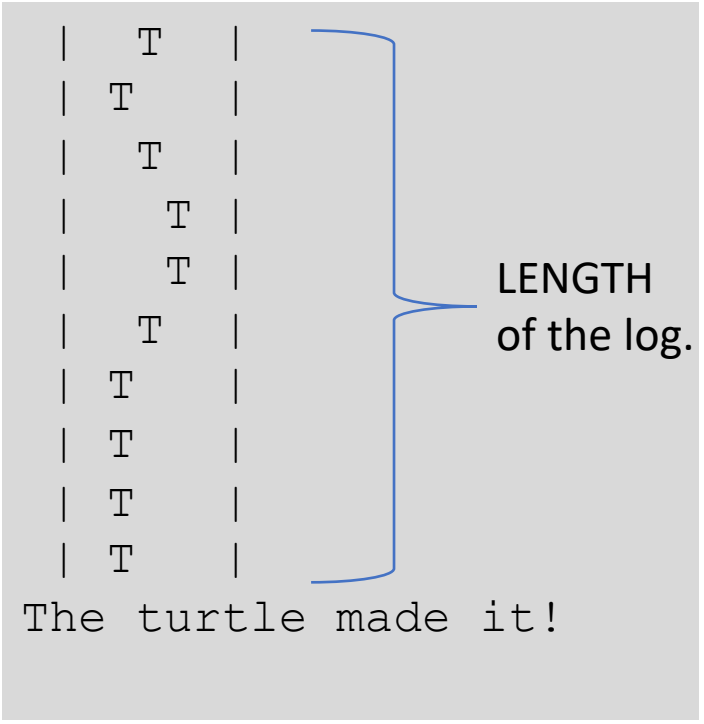
Turtle Crossing a Log

A turtle crossing a log moves one step at a time. Starting in the middle, it randomly takes a step straight, to the left, or to the right.



If the turtle falls off the log before it gets to the other end, it will need to swim the rest of the way. Use constants for the **LENGTH** and **WIDTH** of the log. Print out the turtle's journey.

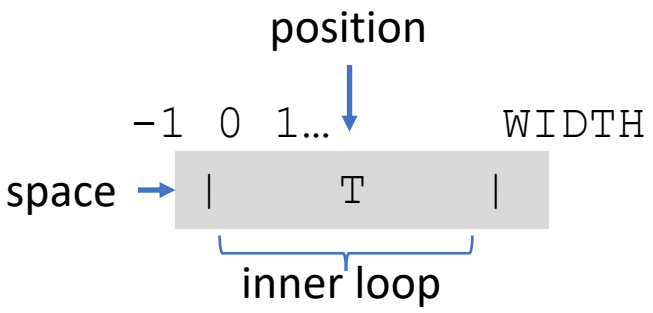


Step 1: Simply print the turtle's position as a number until it falls off:
position < 0 or position >= WIDTH

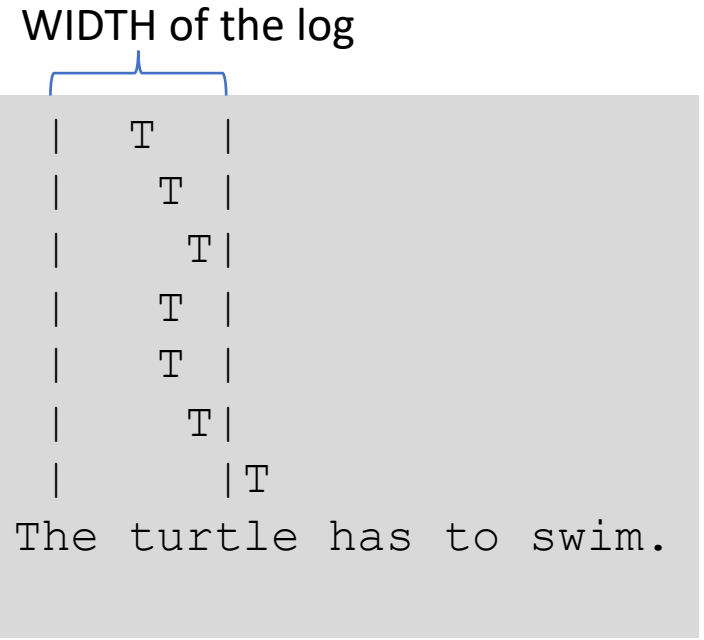
```
Position = 2
Position = 3
Position = 4
Position = 4
Position = 4
Position = 5
```

Step 2: Check if the turtle makes it across.

Step 3: Print the log instead of just the position. Within the outer loop, the turtle should always be on the log



Step 4: After the loop, if the turtle fell, print the log and turtle.



Next steps:

1. Change the value of the constants LENGTH and WIDTH and make sure your program still works. See output on the right for LENGTH=20, WIDTH=15.

2. If you finish, make another Java program which doesn't print the turtle's journey. Instead it simulates the turtle's journey 1,000,000 times and prints how often the turtle is successful.

```
The turtle made it 536294 times out of 1000000 (53.63%).
```

```
  |         T         |
  |         T         |
  |        T          |
  |        T          |
  |       T           |
  |     T             |
  |     T             |
  |     T             |
  |    T              |
  |     T             |
  |      T            |
  |     T             |
  |     T             |
  |      T            |
  |     T             |
  |     T             |
  |     T             |
  |     T             |
  |      T            |
The turtle made it!
```

Step 1: Printing the turtle's location until it falls off of the log.

```
1
2  public class TurtleStep1 {
3
4      public static final int WIDTH = 5;
5      public static final int LENGTH = 10;
6
7      public static void main(String[] args) {
8          //starting halfway between the edges
9          int position = WIDTH/2;
10
11         //keep going while we are still on the log
12         while(position >= 0 && position < WIDTH) {
13             //print position
14             System.out.println("Position = " + position);
15
16             //nest step
17             int step = (int)(Math.random()*3) - 1;
18             position += step;
19         }
20         //print its final position
21         System.out.println("Position = " + position);
22     }
23
24 }
```

Repeat as long as it is still on the log.

Determine its next step. Pick -1, 0, or 1 at random.

Print the turtle's final position off of the log

Step 2: Check if the turtle made it to the other end.

```
1
2  public class TurtleStep2 {
3
4      public static final int WIDTH = 5;
5      public static final int LENGTH = 10;
6
7      public static void main(String[] args) {
8          //starting halfway between the edges
9          int position = WIDTH/2;
10         int distance = 0;
11
12         //keep going while we are still on the log
13         while(distance < LENGTH && position >= 0 && position < WIDTH) {
14             //print position
15             System.out.println("Position = " + position);
16
17             //nest step
18             int step = (int)(Math.random()*3) - 1;
19             position += step;
20             distance++;
21         }
22         //print its final position
23         System.out.println("Position = " + position);
24         //how did it end?
25         if(distance >= LENGTH)
26             System.out.println("The turtle made it!");
27         else
28             System.out.println("The turtle had to swim.");
29     }
30
31 }
```

Keep track of the distance it travelled.

Update the distance.

Check how it ended.

Print the log picture: 3 solutions.

```
public static final char TURTLE = 'T';
public static final char WALL = '|';
```

1.
```
//print this step
System.out.print(' ');
System.out.print(WALL);

//print the spaces before the turtle
for(int i = 0; i < position; i++) {
    System.out.print(' ');
}
//print the turtle
System.out.print(TURTLE);

//print the spaces after the turtle
for(int i = position+1; i < WIDTH; i++) {
    System.out.print(' ');
}

System.out.println(WALL);
```

2.
```
//print this step
System.out.print(' ');
System.out.print(WALL);

for(int i = 0; i < WIDTH; i++) {
    if(i == position) {
        System.out.print(TURTLE);
    }
    else {
        System.out.print(' ');
    }
}

System.out.println(WALL);
```

3.
```
//print this step
System.out.print(' ');
System.out.print(WALL);

for(int i = 0; i < WIDTH; i++) {
    System.out.print(i != position?' ':TURTLE);
}

System.out.println(WALL);
```

Print the final log picture after the loop.

```
//After the walk, print the final picture.
if(distance >= LENGTH) {
    //made it output
    //we don't print the log, since it now on land
    System.out.println("The turtle made it!");
}
else {
    //swim output - print the last step
    //the first character will be the turtle or a space.
    if(position < 0)
        System.out.print(TURTLE);
    else
        System.out.print(' ');

    //print the log. Note, no turtle, all spaces.
    System.out.print(WALL);
    for(int i = 0; i < WIDTH; i++) {
        System.out.print(' ');
    }
    System.out.print(WALL);

    //print the turtle if it feel off this end.
    if(position >= WIDTH)
        System.out.print(TURTLE);

    System.out.println();

    System.out.println("The turtle has to swim.");
}
```

The turtle is on land, so we don’t need to print the log.

The first spot is either the turtle or a space

Next, the log with no turtle.

Finally, the turtle if it wasn’t printed before.

Next Steps #2: running a ton of turtle trials.

```
public static final int WIDTH = 5;
public static final int LENGTH = 10;
public static final int TRIALS = 1000000;

public static void main(String[] args) {
    int madeItCount = 0;

    for(int i = 0; i < TRIALS; i++) {

        //possible positions are 0, 1, 2, 3, and 4
        int position = WIDTH/2;
        int distance = 0;

        while(distance < LENGTH && position >= 0 && position < WIDTH) {
            //next step
            int step = (int)(Math.random()*3) - 1;
            position += step;
            distance++;
        } //end while

        if(distance >= LENGTH) {
            //made it
            madeItCount++;
        } //end if

    }//end for

    //print result.
    System.out.printf("The turtle made it %d times out of %d (%.2f%%). ",
            madeItCount, TRIALS, (100.0*madeItCount/TRIALS));

}//end main
```

Be sure to reset the distance each time.

Just do the steps, don't output anything.

Keep track of how many times the turtle was successful.

New: print as a method.

```
public static void main(String[] args) {
    //possible positions are 0, 1, 2, 3, ... WIDTH - 1
    int position = WIDTH/2;
    int distance = 0;

    while(distance < LENGTH && position >= 0 && position < WIDTH) {
        printTurtle(position);

        //calculate the next step
        int step = (int)(Math.random()*3) - 1;
        position += step;
        distance++;
    } //end while

    //After the walk, print the final picture.
    if(distance >= LENGTH) {
        //made it output
        System.out.println("The turtle made it!");
    }
    else {
        //swim output - print the last step
        printTurtle(position);

        System.out.println("The turtle has to swim.");
    }
}
```

```
public static void printTurtle(int p) {
    //print this step
    if(p < 0)
        System.out.print(TURTLE);
    else
        System.out.print(' ');

    System.out.print(WALL);

    for(int i = 0; i < WIDTH; i++)
        System.out.print(i != p?' ':TURTLE);

    System.out.print(WALL);

    //print the turtle if it feel off this end.
    if(p >= WIDTH)
        System.out.print(TURTLE);

    System.out.println();
} //end printTurtle
```