

Exercises One (Sept 11): Programming to solve the following 5 problems.

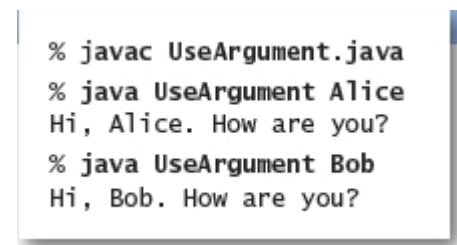
(Hint: Please **DO NOT USE** *selection or loop* statements!)

1.1.6 Modify `UseArgument.java` to make a program `UseThree.java` that takes three names as command-line arguments and prints a proper sentence with the names in the reverse of the order given, so that, for example,

`java UseThree Alice Bob Carol`

prints `Hi Carol, Bob, and Alice.`

```
public class UseArgument {    // UseArgument.java
    public static void main (String[] args) {
        System.out.println( "Hi, " + args[0] + ". How are you?" );
    }
}
```

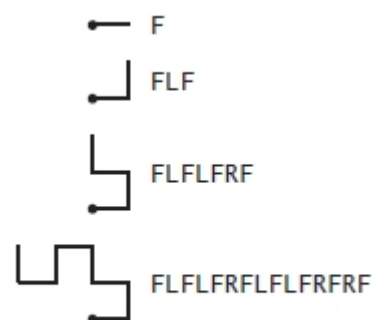


1.2.28 *Order check.* Write a program that takes three double command-line arguments x , y , and z and prints true if the values are strictly ascending or descending ($x < y < z$ or $x > y > z$), and false otherwise.

1.2.30 *Uniform random numbers.* Write a program that prints five uniform random numbers between 0 and 1, their average value, and their minimum and maximum values. Use `Math.random()`, `Math.min()`, and `Math.max()`.

1.2.34 *Three-sort.* Write a program that takes three integer command-line arguments and prints them in ascending order. Use `Math.min()` and `Math.max()`.

1.2.35 *Dragon curves.* Write a program to print the instructions for drawing the dragon curves of order 0 through 5. The instructions are strings of F, L, and R characters, where F means “draw line while moving 1 unit forward,” L means “turn left,” and R means “turn right.” A dragon curve of order n is formed when you fold a strip of paper in half n times, then unfold to right angles. The key to solving this problem is to note that a curve of order n is a curve of order $n-1$ followed by an L followed by a curve of order $n-1$ traversed in reverse order, and then to figure out a similar description for the reverse curve.



Dragon curves of order 0, 1, 2, and 3