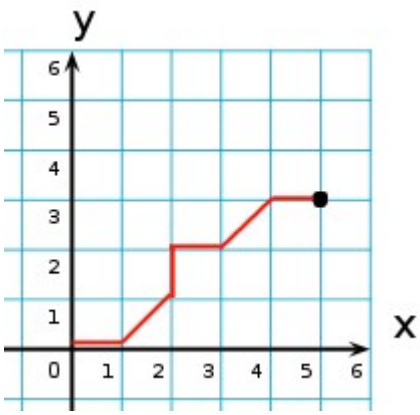
travel

Language/Type: Java recursion recursive backtracking
Author: Marty Stepp (on 2011/02/17)

Write a method `travel` that accepts integers x and y as parameters and uses recursive backtracking to print all solutions for traveling in the 2-D plane from (0, 0) to (x, y) by repeatedly using one of three moves:

- East (E): move right 1 (increase x)
- North (N): move up 1 (increase y)
- Northeast (NE): move up 1 and right 1 (increase both x and y)

The following diagram shows one such path to the point (5, 3).



You may assume that the x/y values passed are non-negative. If x and y are both 0, print a blank line.

The table below shows several calls to your method and the lines of output. Your lines can appear in any order; our output shown tries the possibilities in the order listed above: East, then North, then Northeast.

Call	Output	Call	Output

`travel(1, 2);`	`E N N`
	`N E N`
	`N N E`
		`travel(2, 2);`	`E E N N`
			`E N E N`
			`E N N E`

	N NE NE N		E N NE E NE N N E E N N E N E N E NE N N E E N NE E NE E N NE N E NE NE
travel(2, 1);	E E N E N E E NE N E E NE E		
travel(1, 1);	E N N E NE		

Hint: It may help to define a private helper method that accepts different parameters than the original method. In particular, consider building up a set of characters as a String for eventual printing. Do not use any loops in solving this problem.

Type your solution here:

```java
public void travel(int x, int y) {
    travel(x,y,"");
}

private void travel(int x, int y, String path) {
    if(x == 0 && y == 0)
        System.out.println(path);
    else if(x < 0 || y < 0) {
       // abandon this path
    }
    else {
        travel(x-1,y, path + "E ");     // came from EAST
        travel(x,y-1, path + "N ");     // came from NORTH
        travel(x-1,y-1, path + "NE ");  // came from NORTHEAST
    }
}
```

This is a **method problem.** Write a Java method as described. Do not write a complete program or class; just the method(s) above.

4 Indent

☑ Sound F/X
☑ Highlighting

Submit

You passed 7 of 7 tests.

Go to the next problem: waysToClimb

test #1: `travel(2, 1);`

console output:
```
E E N
E N E
E NE
N E E
NE E
```

result: pass

test #2: `travel(1, 1);`

console output:
```
E N
N E
NE
```

result: pass

test #3: `travel(1, 3);`

console output:
```
E N N N
N E N N
N N E N
N N N E
N N NE
N NE N
NE N N
```

result: pass

test #4: `travel(0, 4);`

console output:
```
N N N N
```

result: pass

test #5: `travel(7, 0);`

console output:
```
E E E E E E E
```

result: pass

test #6: `travel(0, 0);`

console output:

result: pass

test #7: `travel(4, 3);`

console output:
```
E E E E N N N
E E E N E N N
E E E N N E N
E E E N N N E
E E E N N NE
```