

Recursive method tracing
Stack based approach to
2009 AP CS A Multiple Choice #40

By: Brandon Horn
apcstutoring.com

Method `goAgain` makes only 1 recursive call so there is no need to track line numbers.
The initial call is `goAgain("today", 1)` which can be abbreviated as `g("today", 1)`.

```
g("today", 1)
```

`g("today", 1)` calls `g("oday", 2)`.

`g("oday", 2)`

`g("today", 1)`

`g("oday", 1)` calls `g("ay", 3)`.

`g("ay", 3)`

`g("oday", 2)`

`g("today", 1)`

`g("ay", 3)` stops at the base case because $3 \geq 2$.
The method returns its parameter, "ay".

```
g("ay", 3)           returns "ay"  
g("oday", 2)  
g("today", 1)
```

`g("oday", 2)` returns its parameter, "oday",
plus the return value from `g("ay", 3)`.

~~`g("ay", 3)`~~

returns "ay"

~~`g("oday", 2)`~~

returns "odayay"

`g("today", 1)`

`g("today", 2)` returns its parameter, "today",
plus the return value from `g("oday", 2)`.

~~`g("ay", 3)`~~

returns "ay"

~~`g("oday", 2)`~~

returns "odayay"

~~`g("today", 1)`~~

returns "todayodayay"