# Background

In HW3, you are implementing a class that represents large natural numbers as a sequence of digits in a chosen base. The class will allow you to perform arithmetic on numbers with the *same* base, and it will also allow you to convert a number in one base into another base.

The code we will review in this problem will be used in the constructor for class. It takes an `int` value and base and returns that represent that number in that base.

```
public static int[] getDigits(int base, int value)
```

First, we need to define exactly what array of integers the method should return. If we denote this array by "D", then the correct answer must satisfy the equation:

$$value = D[0] + D[1] * base + D[2] * base^2 + \ldots + D[n-1] * base^{n-1}$$

where n is the length of D, and satisfy the requirements that $0 \leq D[0] < base$, $0 \leq D[1] < base$, .., $0 \leq D[n-1] < base$. The two constraints are the *definition* of what it means for the digits in D to represent this value in this base. (They would, in fact, define D uniquely if we added the requirement that D not have any extra zeros at the end, although we will not do that.)

For example, if base = 10 and D = [7, 2, 1], then

$$D[0] + D[1] * base + D[2] * base^2 = 7 + 2 * 10 + 1 * 10^2 = 7 + 20 + 100 = 127$$

The two conditions above (only slightly restated) become the postcondition of `getDigits` on the next page. We will verify the correctness of that implementation.

**Hint**: You will need the following fact about integers, the so-called Division Theorem: if n and m are non-negative integers with m > 0, then n = (n / m) * m + (n % m) and $0 \leq n \% m < m$. (Note: the Java operators "/" and "%" are normally called "div" and "mod" in mathematics.)

# Verifying Correctness

Fill in the missing assertions by reasoning in the direction indicated by the arrows. In each place where two assertions appear next to each other with no code in between (where "?"s appear), provide an explanation of why the top assertion implies the bottom one *on a separate page*.

Notation: we will use "D" for `digits`, "n" for `digits.length`, and "b" for `base`.

{{ Pre: b > 1 and value > 0 and D[0] = D[1] = … = D[n-1] = 0 }}

↓ `int i = 0;`

{{ _____ }}

↓ `int r = value;`

{{ _____ }}

? answer 1 on a separate page

{{ Inv: $0 \le D[0] < b$, $0 \le D[1] < b$, .., $0 \le D[i-1] < b$ and D[i] = … = D[n-1] = 0 and
value = $D[0] + D[1]\, b + D[2]\, b^2 + \ldots + D[i-1]\, b^{i-1} + r\, b^i$ }}

```
while (r != 0) {
```

↓

{{ _____ }}

? answer 2 on a separate page

{{ _____ }}

↑ `digits[i] = r % base;`

{{ _____ }}

↑ `r = r / base;`

{{ _____ }}

↑ `i = i + 1;`

{{ Inv }}

↑

`}`

↓

{{ _____ }}

? answer 3 on a separate page

{{ Post: $0 \le D[0] < b$, $0 \le D[1] < b$, .., $0 \le D[n-1] < b$ and
value = $D[0] + D[1]\, b + D[2]\, b^2 + \ldots + D[n-1]\, b^{n-1}$ }}

## Explanations

Try to keep your explanations short and concise. We recommend that you use one sentence per fact in the bottom assertion to justify why that fact holds given the top assertion. Overall, you should need ~1-4 sentences for each explanation.