Exercise 2

In this problem you will think about how contracts and subclassing interact. In order not to make things too complicated, we will work with a very simple mathematical operation, namely computing the remainder of a number v to base b. You might remember from high school that we can write any v as $b * i + r$, where $0 \leq r < b$. For example, when we consider base 2, we can write $5 = 2 * 2 + 1$. If we think base 10, then $42 = 10 * 4 + 2$, and similarly, at base 16, we can write $50 = 16 * 3 + 2$. If we know what b, v, and i are it is particularly easy to compute r, in the following way

$$r = v - b * i.$$

Consider the following class that we give you. Pre and post conditions are spelled out as a comment to the class `Remainder`.

```
class Remainder {
    /* Contract:
          PRE : b > 0, b * i <= v < b * (i+1)
          POST: 0<=return<b */
    int rest (int b, int i, int v) {
        return (v - b * i);
    }
};
```

Now, consider the five subclasses `A`, `B`, `C`, `D`, and `E` in turn. For each class decide, if it the subclass is valid, which means, if it also honors the contract. If you find that a subclass is valid, argue why. Not more than a few logical steps are necessary in those cases. If you find that a subclass is not valid, explain why. Those explanations are best given by a counter example, such that you show for what numbers b, i, or v a pre condition or post condition is violated.

A.
```
class A extends Remainder {
        int rest (int b, int i, int v) {
            return (super.rest (b, i, v) / 4);
        }
    }
```

B.
```
class B extends Remainder {
        int rest (int b, int i, int v) {
            return (super.rest (b, i+1, v));
        }
    }
```

C.
```
class C extends Remainder {
        int rest (int b, int i, int v) {
            return (super.rest (b / 2, i, v));
        }
    }
```

D.
```
class D extends Remainder {
        int rest (int b, int i, int v) {
            return (2 * super.rest (b, i, v));
```

```
         }
     }

E. class E extends Remainder {
         int rest (int b, int i, int v) {
             if (super.rest(b,i,v) <= 1) {
                 return 0;
             }
             else {
                 return 1;
             }
         }
     }
```