

Q1.

```
/** = number of digits for n, with no leading 0's. */
public static int length(int n) {
    if (n == 0)
        return n;
    return 1 + length(n/10);
}
```

Q2. (a) */** Constructor: an instance for n. */*

```
public BigInt(int n) {
    sign= (n >= 0 ? 1 : -1);
    n= Math.abs(n);
    numb= new int[length(n)];

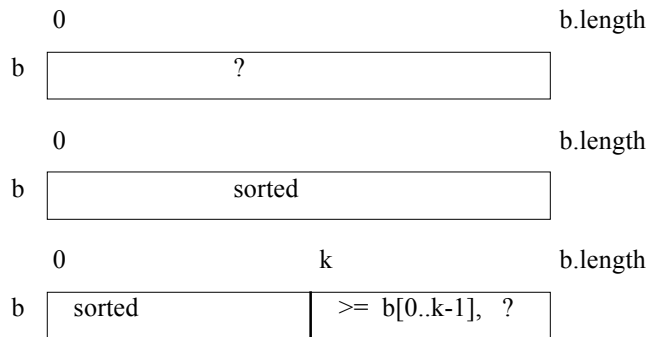
    int nn= n;
    //inv: numb[0..k-1] contains the first k digits of n
    // and the digits of nn belong in numb[k..]
    for (int k= 0; k < numb.length; k= k+1){
        numb[k]= nn%10;
        nn= nn/10;
    }
}
```

(b) It is legal to put **super()**; as the first statement of the constructor; it would call the constructor of superclass Object. Since we did not put that statement in, DrJava inserts it anyway, as the default.

(c) */** Constructor: an instance with value 0 */*

```
public BigInt() {
    this(0);
}
```

Q3. (a) pre- and post-conditions and invariant:

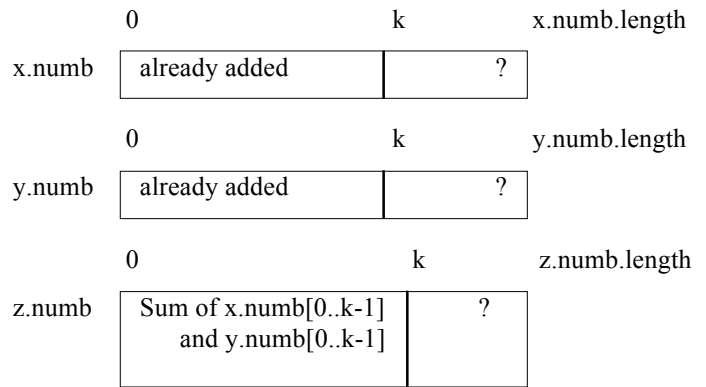


```
for (int k= 0; k < b.length; k= k+1) {
    Set j to index of minimum of b[k..b.length-1];
    Swap b[k] and b[j]
}
```

```
(b) //Set j to index of minimum of b[k..b.length-1]
int j= min(b, k, b.length-1);

//Swap b[k] and b[j]
BigInt t= b[k];
b[k]= b[j];
b[j]= t;
```

Q4. (a) invariant



```
(b) carry= 0;
// invariant: as pictured above
for (int k= 0; k < z.numb.length; k= k+1) {
    int j= carry +
        (k < x.numb.length ? x.numb[k] : 0) +
        (k < y.numb.length ? y.numb[k] : 0);
    carry= j / 10;
    z.numb[k]= j % 10;
}
```

Q5. (a)

```
/** = "b is a BigInt and contains the same integer as
this one" */
public boolean equals(Object b) {
    return b instanceof BigInt &&
        compareTo((BigInt)b) == 0;
}
```

(b) */** = "b is a BigInt and contains the same integer as
this one" */*

```
public boolean equals(Object b) {
    if (!(b instanceof BigInt))
        return false;
    BigInt bb= (BigInt) b;
    if (sign != bb.sign || numb.length != bb.numb.length)
        return false;

    // return false if numb[k] != bb.numb[k] for some k
    // inv: numb[0..k-1] = bb.numb[0..k-1]
    for (int k= 0; k != numb.length; k= k+1) {
        if (numb[k] != bb.numb[k])
            return false;
    }
    return true;
}
```