

Foundations of Programming (Java) 2000

p10q2
fthk

The required constructor is:

```
private BigNo(int n, BigNo b)
{ this.dig = n;
  this.rest = b;
}
```

The two constructors serve different purposes. The one shown in the question is accessed from outside the object within which it is declared and converts an ordinary int into a BigNo. This has to be public. The other constructor is accessed from add() methods when the length of a BigNo is to be increased by one digit. Since it is accessed from within the object it may be declared private as shown.

[4 marks]

The required add methods are:

```
private BigNo add(int c)
{ int d = this.dig+c;
  if (this.rest == null)
    return new BigNo(d);
  else
    return new BigNo(d%10, this.rest.add(d/10));
}

public BigNo add(BigNo that)
{ return this.add(that, 0);
}
```

The three add() methods serve different purposes. Only one is accessed from outside the object within which it is declared and therefore has to be public. Its purpose is to add one BigNo to another as in the call jack.add(jill).

In general the add() method will also need to take a carry digit into account and will thus need an extra argument as in the add() method shown in the question. Users will not need to be concerned with carry digits so the add() method may be private.

The other add() method (the first shown above) is concerned with the special case of adding a carry digit but no associated BigNo. This version is accessed only internally and so may be private.

[6 marks]

A suitable toString() method is:

```
public String toString()
{ return (this.rest == null ? "" : this.rest.toString()) + this.dig;
}
```

[4 marks]

The effect of the call jack.add(jill) is best described by the following annotated diagrams...

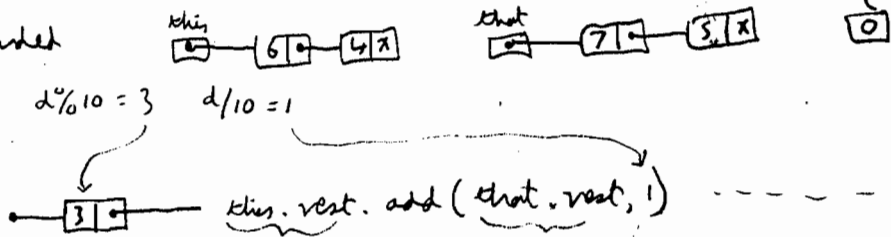
[6 marks]

Main add() method is invoked

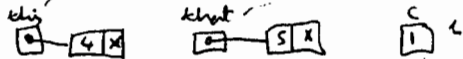
Set, $d = 6 + 7 + 0$

$d \% 10 = 3$ $d / 10 = 1$

Returns a new BigNo:



Main add() now invoked



Since this.rest == null

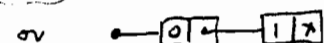
the method returns that.add(4+1)

First add() method is invoked

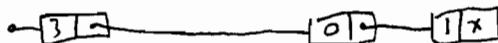


and evaluates $d = 5 + 5$

Since this.rest == null the method returns new BigNo(10)



Net result



or 103

A complete test program which exercises class BigNo is shown below.
Candidates are NOT required to write this out.

```
public class BigNumbers
{
    public static void main(String[] args)
    {
        BigNo jack = new BigNo(46);
        BigNo jill = new BigNo(57);
        System.out.println(jack.add(jill));
    }
}

class BigNo
{
    private int dig;
    private BigNo rest;

    public BigNo(int n)
    {
        this.dig = n%10;;
        if (n/10 == 0)
            this.rest = null;
        else
            this.rest = new BigNo(n/10);
    }

    private BigNo(int n, BigNo b)
    {
        this.dig = n;
        this.rest = b;
    }

    private BigNo add(int c)
    {
        int d = this.dig+c;
        if (this.rest == null)
            return new BigNo(d);
        else
            return new BigNo(d%10, this.rest.add(d/10));
    }

    public BigNo add(BigNo that)
    {
        return this.add(that, 0);
    }

    private BigNo add(BigNo that, int c)
    {
        if (this.rest == null)
            return that.add(this.dig+c);
        if (that.rest == null)
            return this.add(that.dig+c);
        int d = this.dig + that.dig + c;
        return new BigNo(d%10, this.rest.add(that.rest,d/10));
    }

    public String toString()
    {
        return (this.rest == null ? "" : this.rest.toString()) + this.dig;
    }
}
```