Tentamen – extra del: lösning

Uppgifter: lösningar

Uppgift 1 (4 poäng)

```
// sort sorterar element i en vektor enligt deras storlek,
// från och med det minsta till och med det största
public static void sort (int[] element)
{
    int    e = 0;
    int    halPos = 0;
    for (int pos = 1; pos < element.length; pos++)
    {
        e = element[pos];
        halPos = pos;
        while (halPos > 0  &&  e < element[halPos - 1])
        {
            element[halPos] = element[halPos - 1];
            halPos--;
        }

        element[halPos] = e;
     }
}
```

Uppgift 2 (4 poäng)

```
public static String calculate (String n1, String n2)
{
    BigInteger    m = new BigInteger (n1);
    BigInteger    n = new BigInteger (n2);

    BigInteger[]   pq = m.divideAndRemainder (n);
    BigInteger    p = pq[0];
    BigInteger    q = pq[1];
    BigInteger    r = p.add (q);
    BigInteger    res = r.pow (2);

    return res.toString ();
}
```

Uppgift 3 (4 poäng)

```
// add lägger till ett givet heltal till listan
public void add (int value)
{
    Node    node = new Node (value);

    if (first == null)
        first = node;
    else
    {
        Node    n = first;
        while (n.next != null)
            n = n.next;
        n.next = node;
     }
}
```

Uppgift 4 (4 poäng)

```
class Int implements Comparable<Int>
{
    private int    n;

    public Int (int n)
    {
        this.n = n;
    }

    public String toString ()
    {
        return "" + n;
    }

    public int compareTo (Int i)
    {
        int    j = 0;
        if (this.n < i.n)
            j = -1;
        else if (this.n > i.n)
            j = 1;

        return j;
    }
}
```

Uppgift 5 (1 poäng + 2 poäng + 1 poäng)

a) (1 poäng)

7, 9, 5, 4, 1

b) (2 poäng)

Element i sekvensen grupperas i par. Det minsta elementet i varje par urskiljs, och på så sätt skapas en delsekvens. Man upprepar samma strategi i samband med delsekvensen. Samma procedur upprepas tills bara ett element återstår.

I denna strategi glömmer man bort att den ursprungliga sekvensen, eller någon av delsekvenserna på vägen, kan innehålla ett udda antal element. Det sista, oparade elementet glöms i så fall bort. Om detta element är det minsta, får man ett felaktigt resultat.

c) (1 poäng)

Om längden av den ursprungliga sekvensen är av formen 2^k, $k = 0, 1, 2, 3, \ldots$, (alltså för sekvenslängderna 1, 2, 4, 8, …), blir inga oparade element på vägen, och man får korrekt svar.