Solution notes

Programming in Java (Section D) 2005 – Paper 1 Question 10 ACN

Here is my code to show that something can be done in a fairly concise way...

```
import java.util.*;

// I will have a set of sets as the things I can reach at present.
// I will keep just exponents ion them so I play addition on exponents
// rather than multiplication.
// I will start off with just {{1}} to represent the state after
// zero multiplications

public class Power
{

public static void main(String []args)
{
    int n = Integer.parseInt(args[0]);
    System.out.printf("Try to solve for n = %d%n", n);
    HashSet<HashSet<Integer>> ways =
       new HashSet<HashSet<Integer>>();
    {   HashSet<Integer> base = new HashSet<Integer>();
        base.add(1);
        ways.add(base);
    }
    for (int mults=1; mults<n; mults++)
    {
        HashSet<HashSet<Integer>> newways =
            new HashSet<HashSet<Integer>>();
// Now to generate a new configuration I take two items from an existing
// possibility and multiply (ie add!) them together.
        for (HashSet<Integer> x : ways)
        {   for (int i : x)
            {   for (int j : x)
                {   if (i > j) continue;
// here we have x a set, and i <= j two values within it. If i+j is
// already present then that would be dull...
                    if (x.contains(i+j)) continue;
// Make a new set that has all of x plus the new item.
                    if (i + j == n)
                    {   System.out.printf("Done it via %d:%d%n", i, j);
                        System.exit(0);
                    }
                    HashSet<Integer> x1 = (HashSet<Integer>)x.clone();
                    x1.add(i+j);
// and add it to the new list of possibilities. The fact that these
// are hashsets should mean that duplicates always get discarded!
```

```
                    newways.add(x1);
                }
            }
        }
// Now I have collected all the new ways...
        ways = newways;
        System.out.printf("After %d multiplications I can get...%n", mults);
        for (HashSet<Integer> h : ways)
        {   for (int i : h)
                System.out.printf("%4d ", i);
            System.out.printf("%n");
        }
    }
}
```