ESC101 : Fundamental of computing

Tutorial sheet 10 **30 October, 2008**

Topics covered : Recursion

1. Explain in complete details the execution of the following program.

```
class recursion_example
{   public static void Strange(int n, String S)
    {
        if(n==0) System.out.println(``[#''+S+'']'');
        else
        {  Strange(n-1,S+'',''+n);
          if(n>1)  Strange(n-2,S+'',''+(n-2));
        }

    }

    public static void main(String args[])
    {
        Strange(4,'''');
    }
}
```

Answer :

```
[#,4,3,2,1]
[#,4,3,0]
[#,4,1,1]
[#,2,2,1]
[#,2,0]
```

2. Given a non-negative integer n, **Partition**(n) is the set of all the permutations of positive integers which sum upto n. For example, **Partition**(3)=$\{[3],[2,1],[1,2],[1,1,1]\}$. We want to write a program to print all elements of the set **Partition**(n) for a positive integer n, where for each element, we print the corresponding string. For example, the element $[2,1]$ can be expressed as string "[-2-1-]". For sake of simpler formulation, we extend **Partition**(n) to **PartitionS**(n,S) where S is a string as follows.

   PartitionS(n,S) is the set of strings of the form $S+P$ where P is a string corresponding to partition of n. It is easy to observe that **Partition**(n) = **PartitionS**(n,"")

   Interact with the class closely to come up with the following recursive formulation of **PartitionS**(n,S).

$$\textbf{PartitionS}(n,S) = \begin{cases} \text{"["+S+"-]"} & \text{if } n=0 \\ \bigcup_{1\le i\le n} \textbf{PartitionS}(n-i,\text{S+"-"+i}) & \text{if } n>0 \end{cases}$$

   Based on the above formulation, write a recursive method for enumerating elements of set **Partition**(n) on the monitor.