

ESC101 : Fundamental of computing

Quiz 3(A) **SOLUTION**

6 November, 2008

Name :

Roll no :

Section:

Duration : 20 minutes

NOTE : There are TWO questions, flip the front page to read the second question

Question 1 (marks=5)

What is output of the following program ?

```
class fun1
{
    public static void fun(int n, String S)
    {
        if(n==0) System.out.println("[-"+S+"]");
        else
        {
            fun(n-1,S+", "+n);
            if(n>1) fun(n/2,S+", "+(n-2));
        }
    }

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

Answer : (One marks for each correct output.)

[-,4,3,2,1]

[-,4,3,0,1]

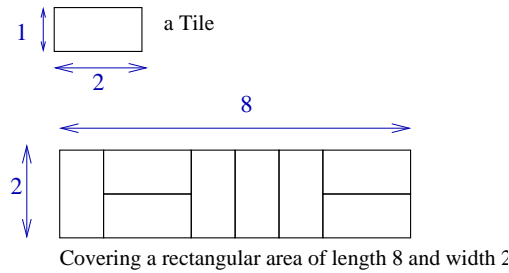
[-,4,1,1]

[-,2,2,1]

[-,2,0,1]

Question 2 (marks=5)

We have a large collections of identical rectangular 2×1 tiles (length of each tile is 2 units and width is 1 unit). We want to cover a given $2 \times n$ rectangular area using these tiles. A tile may be placed horizontally or vertically while covering an area. There may be various possible ways to do so. For example, Figure given below gives one possible covering of a 2×8 rectangular area by the tiles.



You have to write a method `PossibleTilings (int n)` which counts the number of ways to cover a $2 \times n$ rectangular area using the 2×1 tiles. Assume that n takes positive value only.

Answer : Here are two sample solutions, there may be many other correct solutions as well. Two marks are for the base case(s), and three marks are for the recursive case. There is binary marking for both these parts. One will get two marks for the base case if his/her base case is fully correct and zero otherwise. One one will get three marks for the recursive case if the recursive part is fully correct, and zero otherwise.

```
public static int PossibleTilings(int n)
{
    if(n<2) return 1;

    else return (PossibleTilings(n-1) + PossibleTilings(n-2));
}
```

OR

```
public static int PossibleTilings(int n)
{
    if(n==1) return 1;

    else if(n==2) return 2;

    else return (PossibleTilings(n-1) + PossibleTilings(n-2));
}
```