Derangements

ZPRAC-16-17-Lab9

Derangements [30 points]

ANNOUNCEMENT:
Up to 20% marks will be allotted for good programming practice. These include
- Comments for non trivial code
- Indentation: align your code properly

Up to 50% marks can be deducted if you do not use recursion

Use long long int data type to store integers for this problem. Do not worry about overflow issues.

A derangement is a permutation in which none of the objects appear in their "natural" place. For example, the only derangements of {1,2,3} are {2,3,1} and {3,1,2}. The number of derangements for n distinct objects has a well known recursive formulation defined as follows:

$D(n)=$

- 0, if $n<2$
- 1, otherwise if n = 2

$(n-1)*(D(n-1)+D(n-2))$, otherwise

In this problem, you have to compute $Derangement(n)$ for the given n and count the number of recursive calls. Each call to the function Derangement() should be considered as a separate call.

NOTE: The derangement function should be defined using recursion only (hence use of arrays is not allowed). Follow the exact recurrence given above, otherwise you may face issues in the count-of-recursive-calls.

Input Format:
The first line of input is a number t which indicates the number of test cases. Then, t lines

follow where each line contains n.

Output Format:

Print $Derangement(n)$ and count-of-recursive-calls separated by a space, one per line for each test case.

EXAMPLE:

Input:

3
1
2
3

Output:

0 1
1 1
2 3