

# Problem 3

When I first learned backtracking I made a program to find all the permutations of the English alphabets in lexicographically increasing. Filled with elation I showed the program to Rohil. Rohil being someone who likes to do stuff off the league was not impressed and gave me the following variation of the problem help me to solve the problem:

You have to find the number of permutations of length  $N$  ( $1 \leq N \leq 11$ ) such that at whenever an alphabet (say 'c') appears in the permutation all the alphabets smaller than 'c' should have appeared before it at least once. An alphabet is smaller than another if it appears before the other in the English alphabet. 'a' being the smallest and 'z' being the largest. For example when  $N=2$  then aa,ab are the only valid permutations and ba,bb is invalid since in ba all the alphabets smaller than b have not appeared at least once before it. See example for further clarification.

## Input

Line 1: T(no. of test cases)

Line 2: n1

Line 3: n2

...

...

## Output

Line 1: no. of such permutations of length n1

.....

.....

## Example

**Input:**

2  
2  
3

**Output:**

2  
5

Explanation for  $N=3$ , the possible permutations are:

abc  
aba  
abb  
aab

aaa