**Problem :** [**https://practice.geeksforgeeks.org/problems/ncr1019/1#**](https://practice.geeksforgeeks.org/problems/ncr1019/1#)

**Approach1: Using DP. (when n\*r<=10^7)**

nCr = n-1Cr-1 + n-1Cr

(Either select current person and so now need to select n-1 people from r-1,

don’t select current person , then need to select r persons from rest n-1).

**Time = O(n\*r)**

**Approach2: Using Fermat’s little theorem (when mod value is prime).**

-> Using normal formula : n! / (r! \* (n-r)!) , and fermat theorem and binary exponentiation.

-> (imp)Convert to long long when multiplying.

**Time = O(n + logm)** where O(n) is for finding factorials.

**Approach3: Using Lucas Theorem.(when n and r are large)**

Write n and r in base m

n = [ in base m ]

r = [ in base m ]

(Each digit < m)

nCr mod m = (((C \* C ) mod m ) \* C ) mod m ….

int nCrDigits(int n, int r)  
{  
// … Use any method for nCr of digits, as discussed above

// [Pascal's triangle or Modulo inverse method]   
}

int nCr(int n, int r, int m)  
{  
 if(r==0)

return 1;  
 int dig\_n = n%m;

int dig\_r = r%m;

return (nCrDigits(dig\_n,dig\_r,m) \* nCr(n/m, r/m, m))%m;  
}

Time complexity: O(logmn \* Time\_complexity(nCrDigits) )

**Code1 :** [**https://practice.geeksforgeeks.org/viewSol.php?subId=20f47867fac25b59d671a7b29d105df1&pid=704542&user=alaymehta1**](https://practice.geeksforgeeks.org/viewSol.php?subId=20f47867fac25b59d671a7b29d105df1&pid=704542&user=alaymehta1)

**Code 2 :** [**https://practice.geeksforgeeks.org/viewSol.php?subId=20f47867fac25b59d671a7b29d105df1&pid=704542&user=alaymehta1**](https://practice.geeksforgeeks.org/viewSol.php?subId=20f47867fac25b59d671a7b29d105df1&pid=704542&user=alaymehta1)

**2. Compute nPr :**

nCr \* r! = n! / (n-r)!

1. Using formula : n! / (n-r)!
2. **Using DP :**

nPr = n-1Pr + r \*(n-1Pr-1)

-> Not select the current person , so now need to select ‘r’ people from remaining n-1 OR select current person and find permutations of rest r-1 people from remaining n-1, and then we can place current element in any of the ‘r’ remaining gaps.

**n, k) = P(n-1, k) + k\* P(n-1, k-1)**