nCr

```
//tabla para ir guardando cálculos
//(programación dinámica)
unsigned long long dp[Nmax][Rmax];

unsigned long long nCr(int n, int r)
{
    if (n==r) return dp[n][r] = 1;
    if (r==0) return dp[n][r] = 1;
    if (r==1) return dp[n][r] = (unsigned long long)n;
    if (dp[n][r]) return dp[n][r];
    return dp[n][r] = nCr(n-1, r) + nCr(n-1, r-1);
}
```

mcd

```
int mcd(int a, int b)
{
    while(b) b ^= a ^= b ^= a %= b;
    return a;
}
```

Subfactorial

```
ull factorial(int n)
{
     if (n > 0)
          return n * factorial(n-1);
     else return 1;
}

ull subf(int n)
{
     double res = factorial(n) / M_E;
     ull res2 = res + 0.5; //entero más cercano
     return res2;
}
```