$$^{n}C_{r} \Rightarrow \frac{n!}{r!\,(n-r)!}$$

Choosing r items out of n distinct items.

Ways of calculating $^{n}C_{r}$

1) $^{n}C_{r} = {}^{n-1}C_{r} + {}^{n-1}C_{r-1}$

Pascal formulae

	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
2	1	2	1	0	0	0	0
3	1	3	3	1	0	0	0
4	1	4	6	4	1	0	0
5	1	5	10	10	5	1	0
6	1	6	15	20	15	6	1

Code for pascal triangle

(for n numbers [0,n]

```
int nCr [n+1][n+1]
// set everything as 0
for (i=0; i≤n; i++)
    nCr[i][0] = 1

for (i=1; i≤n; i++) {
    for (j=1; j≤i; j++) {
        nCr[i][j] = (nCr[i-1][j]
                    + nCr[i-1][j-1])
                    % M
    }
}
```

TC: $O(n^2)$

SC: $O(n^2)$

what is ${}^{75}C_{48}$

nCr[75][48]

$N \to 10^5$

$150 \% 20 \Rightarrow 10$

2) Pre-calculate the factorials and the inverse of them.

$$^{n}C_{r} \Rightarrow \frac{n!}{r!\,(n-r)!}$$

$$n! \times \underbrace{\frac{1}{r!}}_{\text{inv of } (r!)} \times \frac{1}{(n-r)!} \rightarrow \text{Inv of } (n-r)!$$

int fact[n+1]   fact(i) = i! % M

int invfact[n+1]   invfact(i) = inverse of fact(i)

inverse of a wrt M
= pow(a, M-2, M)

fact[0] = 1
fact[1] = 1

```
for (i=2 ; i≤n ; i++) {
    fact[i] = (i × fact[i-1]) % M
}
```

```
for (i=0 ; i≤n ; i++) {
    invfact[i] = pow(fact[i], M-2, M)
}
```

$a^{M-1} \% M = 1$ M prime

$(a \times a^{M-2}) \% M = 1$

$\left(a \% M \times a^{M-2} \% M\right) \% M = 1$

$[a \times \text{pow}(a, M-2, M)] \% M = 1$

n, r what is ${}^nC_r$

$n! \times invfact(r) \times invfact(n-r)$

$(fact[n] \times invfact[r] \times invfact[n-r]) \% M$

ans = $(fact[n] \times invfact[r]) \% M$

ans = $(ans \times invfact[n-r]) \% M$

$(a\%m \times b\%m \times c\%m) \% m$

$x \% m \implies [0, m-1]$

$[(m-1)(m-1)(m-1)] \% m$

$M = 10^9$

$\approx \boxed{10^{2\cancel{2}}}$

2 at a time $\Rightarrow$ $\boxed{10^{15}}$ long

Hockey Stick Rule

$$ {}^{n}C_0 + {}^{n+1}C_1 + {}^{n+2}C_2 + {}^{n+3}C_3 + \cdots + {}^{n+r}C_r = {}^{n+r+1}C_r $$

$$ {}^{n}C_0 = 1 = {}^{n+1}C_0 $$

$$ {}^{n}C_r = {}^{n-1}C_r + {}^{n-1}C_{r-1} $$

$$ \underbrace{{}^{n+1}C_0 + {}^{n+1}C_1}_{} + {}^{n+2}C_2 + {}^{n+3}C_3 + $$

$$ \cdots + {}^{n+l}C_l = {}^{n+l+1}C_l $$

$$ \underbrace{{}^{n+2}C_1 + {}^{n+2}C_2}_{} + {}^{n+3}C_3 + \cdots $$

$$ \underbrace{{}^{n+3}C_2 + {}^{n+3}C_3}_{} $$

$^{n+4}C_3$

$$\underbrace{{}^{n+r}C_{r-1} + {}^{n+r}C_r}$$

$^{n+r+1}C_r$

```
int nCr ( int n, int r) {
    ans = (fact[n] × invfact[r]) % M
    ans = (ans × invfact[n-r]) % M
}
```

Q1 You are given Q queries of the form n, r. For each query print the number of ways to select either r from n OR r/2 from n/2

Constraints ⇒ Q 10^5

n, r 10^5

```
fact[0] = 1
fact[1] = 1
for (i=2 ; i≤10^5 ; i++) {
    fact[i] = (i × fact[i-1]) % M
}

for (i=0 ; i≤10^5 ; i++) {
    invfact[i] = pow(fact[i], M-2, M)
}

for (i=0 ; i<q ; i++) {
        read(n, r)
```

// ans is $^nC_r + ^{n/2}C_{r/2}$

ans = [nCr (n, r) +
nCr (n/2, r/2)] % M

print (ans)

}

Q2 Given an array of distinct no.s, find out no of ways of choosing K even no.s from the array.

Eg - { 1, 2, 3, 4 } K = 1

ans = 2

N is till 10^5
K is till 10^5

we are only choosing from even no.s.

```
for (i=0 ; i<n ; i++) {
    if ( a[i] % 2 == 0)
        count ++
}
```

choose k from count

ans = nCr(count, k)

$^{count}C_k$

{done}