Factorial

5! = 120

1 * 2 * 3 * 4 * 5

4! = 1 * 2 * 3 * 4 = 24

n! = **product of** all the numbers from 1 to n

1! = 1

0! = 1

Ques. write a function that takes n as input and return factorial of n as answer.

n = 3 → ans 6

```
int factorial ( int n ) {
    int ans = 1 ;
    for(int i=1; i < = n; i++) {
        ans = ans *i;
    }
    return ans;
}
```

n = 3   ans = 1   ans = 6 (3*2*1)

i	i < = n	ans
1	true	1*1 = 1
2	true	1*2 = 2
3	true	2*3 = 6
4	false	→ break

Ques. Calculate the value of $\boxed{{}^nC_r}$

$\boxed{n = 4 \quad r = 2}$

$${}^nC_r = \frac{n!}{(n-r)! * r!}$$

5 fruits types choose any ②

$${}^5C_2 = \frac{5!}{(5-2)! * 2!} = \frac{120}{6 * 2} = 10$$

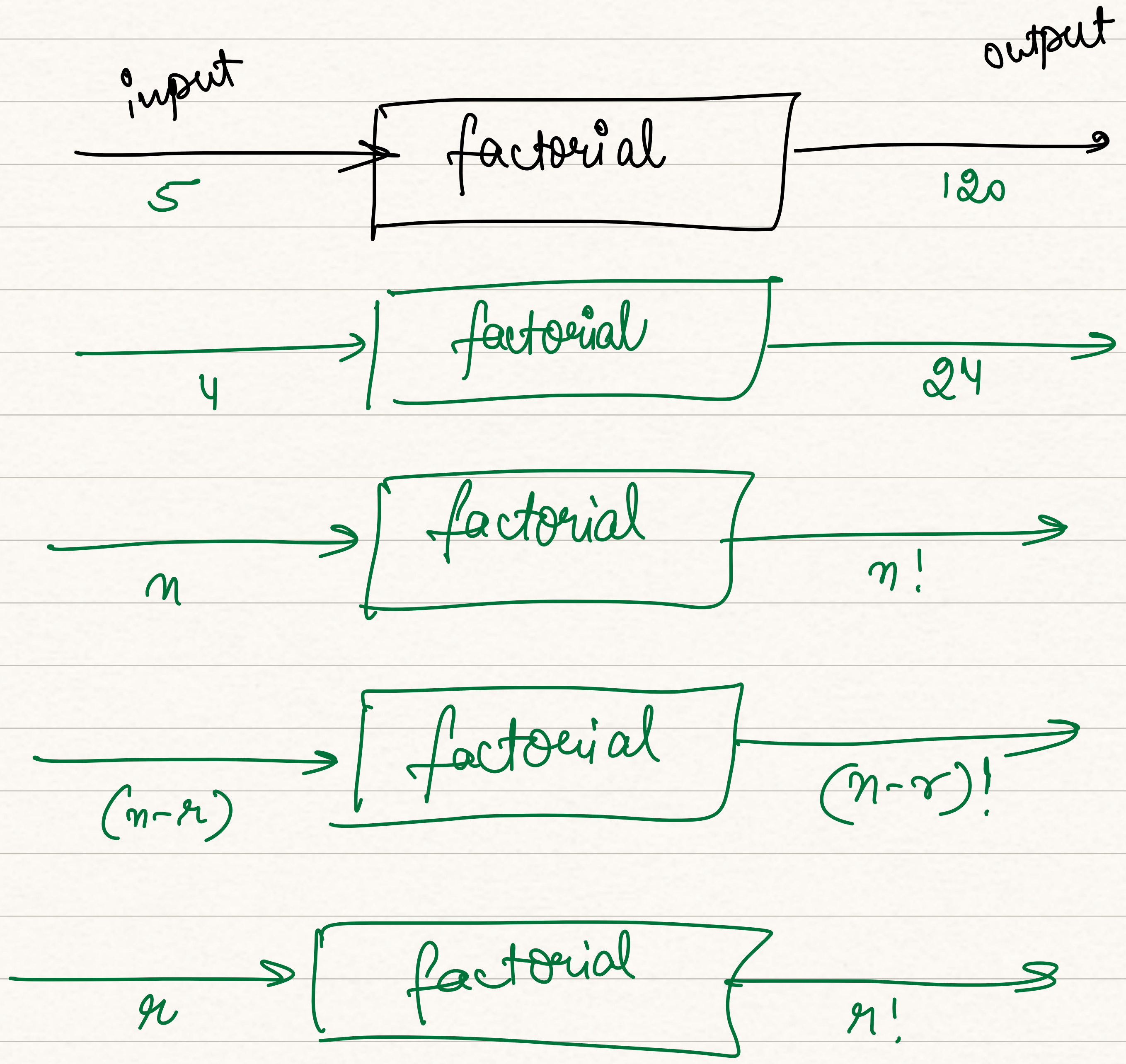
$${}^4C_2 = \frac{4!}{(4-2)! * 2!} = \frac{24}{2 * 2} = 6$$

① find $n!$ → loop 1 to n

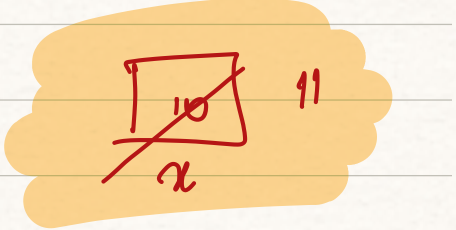
② find $(n-r)!$ → loop 1 to $(n-r)$

③ find $r!$ → loop 1 to r

④ calculate ${}^nC_r$


input
output
factorial
5
120
factorial
4
24
factorial
n
n!
factorial
(n-r)
(n-r)!
factorial
r
r!

```
main () {
    if (true) {
        int x = 10
        S.O.P (x);
        x++;
    }
    S.O.P (x); // cannot find symbol
}
```



Error :

Scope of x is till the if block

Square root of a number

function that will return square root if the given number is a perfect square otherwise return -1.

	ans
$n = 81$	9
$n = 90$	-1

$n = 16$

$i * i == n$ → true

$i =$		
1	$1*1 == 16$	X
2	$2*2 == 16$	X
3	$3*3 == 16$	x
4	$4*4 == 16$	→

return i

n = 23

i =	1	1*1	== 23	x
	2	2*2	== 23	x
	3	3*3	== 23	x
	4	4*4	== 23	x
	5			

$\boxed{5^2}$ == 23

25 > 23

```
for (int i=1 ;   i*i <= n ;  i++) {
        if (i*i == n) {
            // n is a perfect square
            return i ;
        }
}
return -1;
```

Count of Primes -

function that returns count of primes less than or equals to given number.

			ans
$n = 7$	→	2 , 3 , 5, 7	4
$n = 15$	→	2, 3 , 5, 7, 11, 13	6

2 → n

check all the numbers from 2 to n and increase the count if they are prime.

check prime → find factors
if count of factors == 2
then yes

```
boolean  checkPrime  (int n)  {
    int  factors = 0;
    for ( int  i = 1 ;   i <= n ;  i++) {
            if  (n % i == 0) {
                 factors++;
            }

            if  ( factors > 2) {
                 break;
            }
    }

    if  ( factors == 2)  {
            return  true;
    }
    else {
            return  false;
    }
}
```