



Today's agenda

↳ factorial

↳ nC_r & nP_r

↳ functions



AlgoPrep



Q) factorial

↳ Given N , Print factorial of N .

Quiz 1: $\text{fact}(4): 1 * 2 * 3 * 4 = 24$

$$\text{fact}(N) = 1 * 2 * 3 * 4 * \dots * N$$

// Pseudo code

```
public static void main() {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt();  
  
    int ans = 1;  
    for (int i = 1; i <= n; i++) {  
        ans = ans * i;  
    }  
    System.out.println(ans);  
}
```



$n=4$

~~ans = 12226~~
24

```
int ans = 1;
→ for (int i = 1; i <= n; i++) {
    ans = ans * i;
}
System.out.println(ans);
```

i	i <= n
1	t
2	t
3	t
4	t
5	f

↳ exit



AlgoPrep



11 nC_r and nP_r

$$\text{Quiz 2: } {}^5C_2 \rightarrow \frac{5!}{2! \times 3!} = \frac{120}{6 \times 2} = 10$$

$$\text{Quiz 3: } {}^5P_3 \rightarrow \frac{5!}{2!} = \frac{120}{2} = 60$$

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

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

Q) Given n and r , write an algorithm to calculate nC_r .

$$\frac{n!}{r! (n-r)!}$$



AlgoPrep



// Pseudo code

```
P S r main ( ) {  
    Scanner Sch = new Scanner(System.in);  
    int n = Sch.nextInt();  
    int r = Sch.nextInt();  
  
    int nfact = 1;  
    for (int i = 1; i <= n; i++) {  
        nfact = nfact * i;  
    }  
  
    int rfact = 1;  
    for (int i = 1; i <= r; i++) {  
        rfact = rfact * i;  
    }  
  
    int nmrfact = 1;  
    for (int i = 1; i <= n-r; i++) {  
        nmrfact = nmrfact * i;  
    }  
  
    int ans = nfact / (rfact * nmrfact);  
  
    System.out.println(ans);  
}
```



DRY → Do not repeat yourself

↓
function

↳ Screw drivers → open & close nut & bolts.

⇓
buy Screwdrivers

↳ open & close table's nut & bolt → use the previously bought screw driver.



Syntax:

Public Static int name (input) {

SKIP the meaning for now

output type

function name

// Statement 1

// Statement 2

}

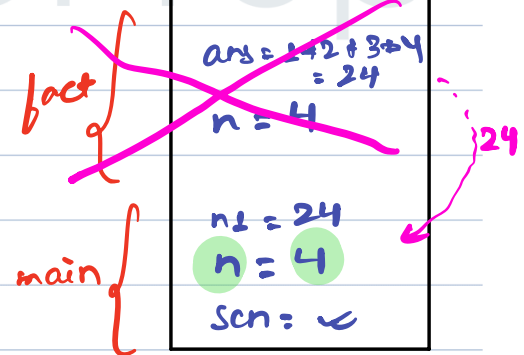


```
Public Static void main() {  
    Scanner Scn = new Scanner(System.in);  
    int n = Scn.nextInt();  
  
    int ans = 1;  
    for (int i = 1; i <= n; i++) {  
        ans = ans * i;  
    }  
    System.out.println(ans);  
}
```

```
Public Static void main() {  
    Scanner Scn = new Scanner(System.in);  
    int n = Scn.nextInt();  
    fact(n);  
}  
  
Public Static int fact(int n) {  
    int ans = 1;  
    for (int i = 1; i <= n; i++) {  
        ans = ans * i;  
    }  
    return ans;  
}
```

Call stack

```
Public Static void main() {  
    Scanner Scn = new Scanner(System.in);  
    int n = Scn.nextInt();  
    → int n1 = fact(n);  
    System.out.println(n1);  
}  
  
Public Static int fact(int n) {  
    int ans = 1;  
    for (int i = 1; i <= n; i++) {  
        ans = ans * i;  
    }  
    return ans;  
}
```



24

return in function == Balak in for loop.

→ Called `func()` but you don't want to return anything from there → type of function should be void.

Balak tu 9:23 PM



// nCr using function

```
public static void main () {  
    Scanner scn = new Scanner (System.in);  
    int n = scn.nextInt();  
    int r = scn.nextInt();  
  
    int nfact = fact (n);  
    int rfact = fact (r);  
    int nmrfact = fact (n-r);  
  
    int ans = nfact / (rfact * nmrfact);  
    System.out.println (ans);  
}
```

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

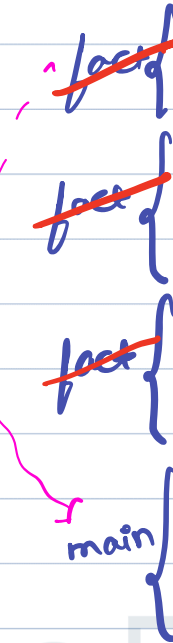

nC_r

Call Stack



```
public static void main() {  
    Scanner sc = new Scanner(System.in);  
    int n = sc.nextInt();  
    int r = sc.nextInt();  
  
    int nfact = fact(n);  
    int rfact = fact(r);  
    int rnmfact = fact(n-r);  
  
    → int ans = nfact / (rfact * rnmfact);  
    System.out.println(ans);  
}
```

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



~~ans = 1 * 1 * 2 = 2~~
~~n = 2~~

~~ans = 1 * 2 * 3 = 6~~
~~n = 3~~

~~ans = 1 * 2 * 3 * 4 * 5 = 120~~
~~n = 5~~

nrmfact = 2
rfact = 6
nfact = 120
r = 3
n = 5
sc = 12

ans = $\frac{120}{12}$
= 10

$(n-r) \Rightarrow 5-3 = 2$

→ main() {

~~int ans = sum(10, 20, 30);~~
Sum(10, 20, 30);
}

int
void sum(int a, int b, int c) {

return a+b+c;

}



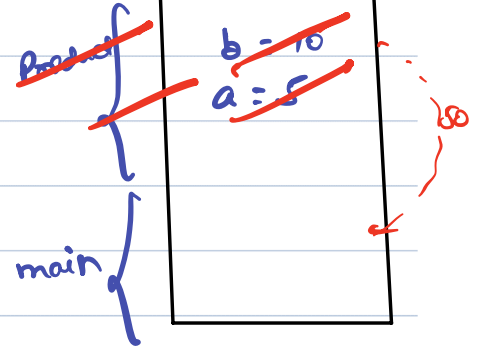
Quiz 4:

```
Public Static void main (String [] args) {  
    Product (5, 10);  
    50  
}
```

→

```
Public Static int Product (int a, int b) {  
    → return a * b;  
}
```

↳ no output



Quiz 5:

```
Public Static void main (String [] args) {  
    int ans = Subtract (5, 10);  
    System.out.println (ans);  
}
```

→ error

```
Public Static void Subtract (int a, int b) {  
    return a - b;  
}
```



Quiz 6:

```
public static void main (String[] args) {  
    int n1 = cube(3);  
    System.out.println (add(n1, cube(2)));  
}
```

6 35

```
public static int add (int a, int b) {  
    return a + b;  
}
```

```
public static int cube (int a) {  
    return a * a * a;  
}
```

