Reusable

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int T = scn.nextInt();
    for (int i = 0; i < T; i++) {
        int x = scn.nextInt();
        int y = scn.nextInt();
        findSum(x, y);
    }
}
// main logic
public static void findSum(int x, int y) {
    int ans = x + y;
    System.out.println(ans);
}
```

o/p

70
5

dry run

T = 2

i = 0, (0 < 2)

x = 30

y = 40

ans = 70

i = 1, (1 < 2)

x = 2

y = 3

ans = 5

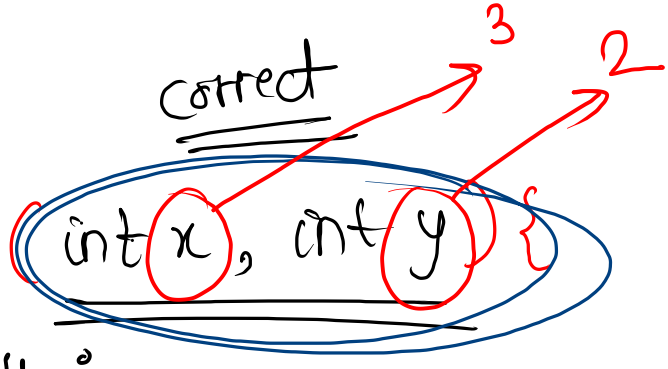
i = 2, (2 < 2) x

Note:-

main

```
int a = 3;
int b = 2;
fun(a, b);
```

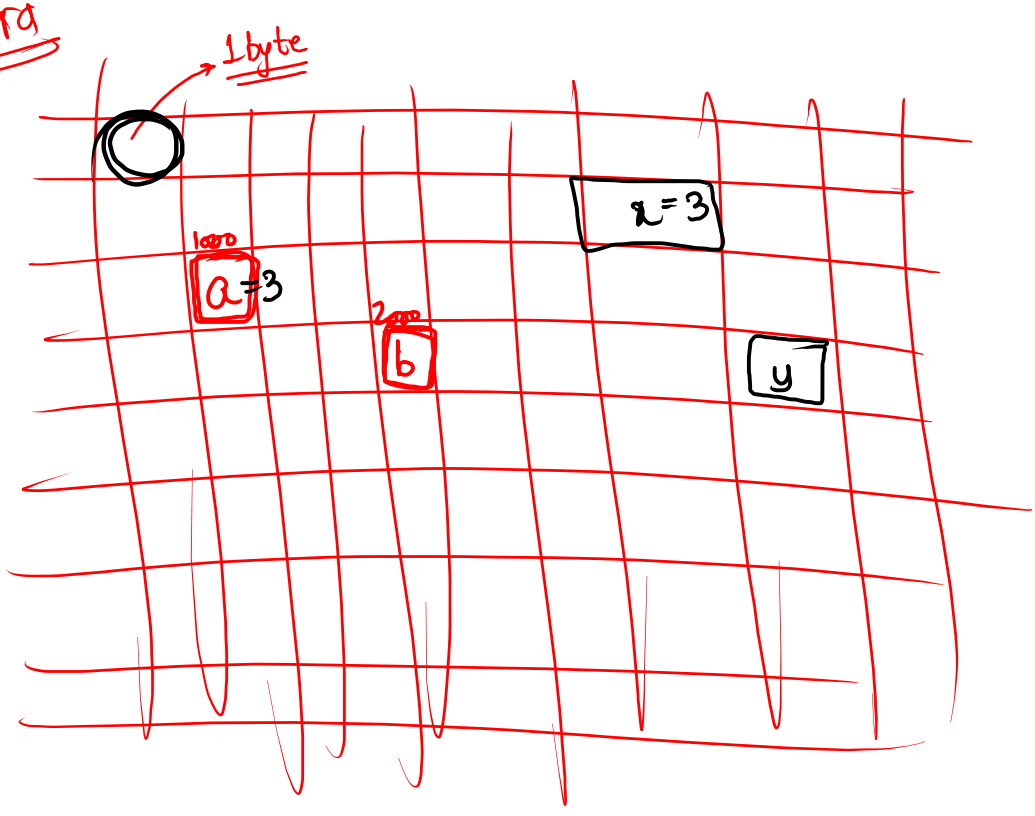
a, b



fun

```
public static void fun
    int sum = x + y;
}
```

Extra



1 Tb
↳ 1024 gb
↳ 1024 * 1024 mb
↳ 1024 * 1024 * 1024 kb
↳ 1024 * 1024 * 1024 * 8 byte

Factorial of N

↳ $n = 5$, $fact = 5 * 4 * 3 * 2 * 1 = 120$

factorial = $n!$

$= n * (n-1) * (n-2) * \ldots * 2 * 1$

Note :- from today, we will use functions in every single qustion.

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    printFactorial(n);
}
// main logic
public static void printFactorial(int n) {
    long ans = 1;
    for (int i = 1; i <= n; i++) {
        ans = ans * i;
    }
    System.out.println(ans);
}
```

Find nCr.

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

combination

$${}^{n}C_{r} = \frac{a}{b * c}$$

where

$a = n!$

$b = r!$

$c = (n-r)!$

$${}^{5}C_{2} = \frac{5!}{2! * (5-2)!}$$

$$= \frac{(5 * \cancel{4}^{2} * 3 * 2 * 1)}{(\cancel{2} * 1) * (3 * 2 * 1)}$$

$$= 10$$

Theory

↳ functions can be divided into 2 categories based on return type

1) Return Type :- returns something as a result
(int, boolean, String, ...)

2) Non-return type; never returns anything
(void)

Ex:-

```
public static void main(String[] args) {
    fun1(); // non return type
    String ans = fun2(); // return type
    System.out.println(ans);
}
public static void fun1(){
    System.out.println("Hello");
}
public static String fun2() {
    return "Hello";
}
```

Note:- any question can solved using both type of functions

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int r = scn.nextInt();
    findCombination(n, r);
}

public static void findCombination(int n, int r) {
    long a = printFactorial(n);
    long b = printFactorial(r);
    long c = printFactorial(n - r);
    long ans = a / (b * c);
    System.out.println(ans);
}

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

5 (5) (2)
2
3
5
2
3

Note:-

break :- terminates the loop

return :- 1) terminates the function

2) returns some value back

Note:- we can't write any statement after return got executed

Find product of the two numbers using function.

non-return type

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int T = scn.nextInt();
    for (int i = 0; i < T; i++) {
        int x = scn.nextInt();
        int y = scn.nextInt();
        findSum(x, y);
    }
}
// main logic
public static void findSum(int x, int y) {
    int ans = x * y;
    System.out.println(ans);
}
```

return type

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int T = scn.nextInt();
    for (int i = 0; i < T; i++) {
        int x = scn.nextInt();
        int y = scn.nextInt();
        int ans = findSum(x, y);
        System.out.println(ans);
    }
}
// main logic
public static int findSum(int x, int y) {
    return x * y;
}
```

Swap x and y

```
int x = 5;
int y = 6;
```

x = 6
y = 5

steps

```
int temp = x
x = y
y = temp
```

x = 5	y = 6	temp
		5
6	6	5
6	5	5

Vo gmp

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int x = scn.nextInt();
    int y = scn.nextInt();
    swap(x, y);
}
public static void swap(int x, int y) {
    int temp = x;
    System.out.println("c = " + x);
    x = y;
    System.out.println("x = " + x);
    y = temp;
    System.out.println("y = " + y);

    System.out.println("x = " + x);
    System.out.println("y = " + y);
}
```