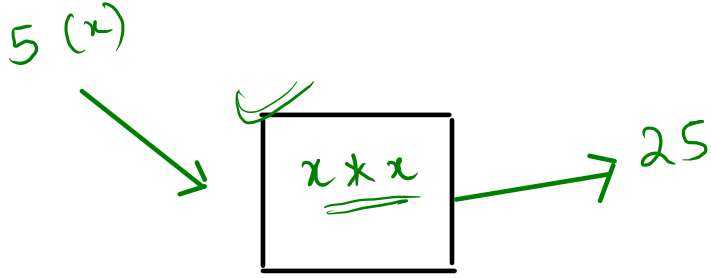
Revision.

- Functions.
  - ? (dry run)
  - Patterns.
    - Tribonacci
      - ?

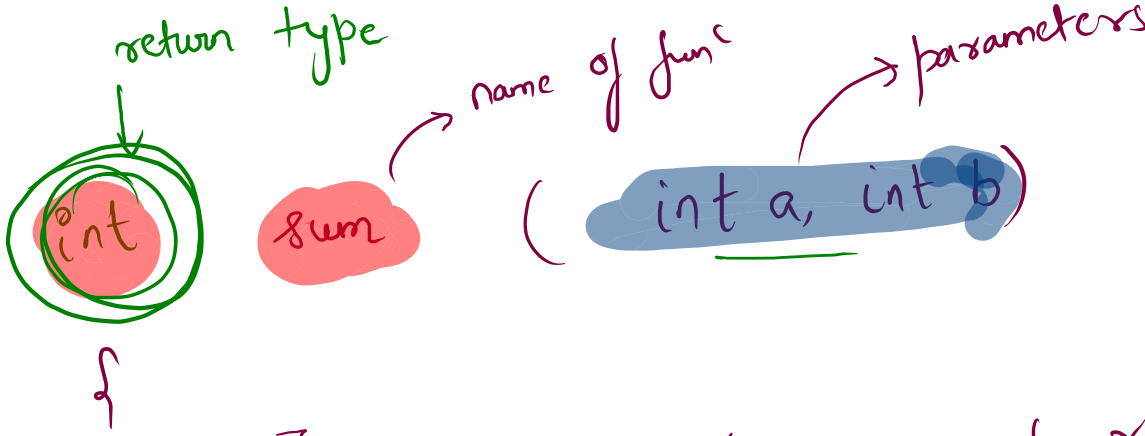
function.

↳ block of code.

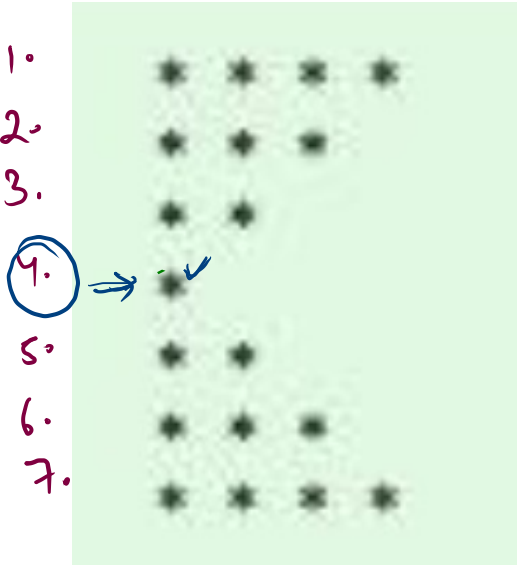
↳ which performs any particular task.



Syntax.


return type
name of func
parameters
int
sum
int a, int b
{
}
body / logic of ur function.

Pattern.



n = 4

rows? = 2n - 1

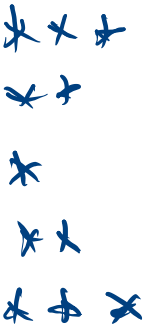
stars = n

after each row.

row < n | star--

row ≥ n | star++ ?

n = 3



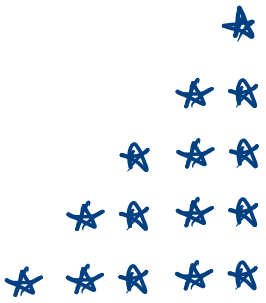
Pattern eg.

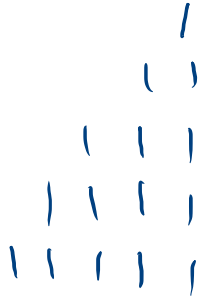
7

Sample Output 0

```
                7
              7 14
            7 14 21
          7 14 21 28
       7 14 21 28 35
 7 14 21 28 35 42
```

basic



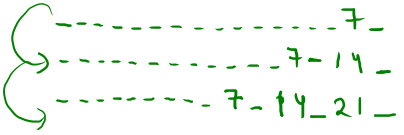


```
15
16      int space= (n-1);
17      int  str=1;
18
19      for(int i=1; i<=n; i++)
20      {
21          for(int spc=1; spc<=space; spc++)
22          {
23              System.out.print("  ");
24          }
25          for(int j=1; j<=str; j++)
26          {
27              System.out.print(k*j + " ");
28          }
29
30          str++;
31          space--;
32          System.out.println();
33      }
34
35
```

space = 5 ✗ 3

str = 1 ✗ 3

n=6
k=7



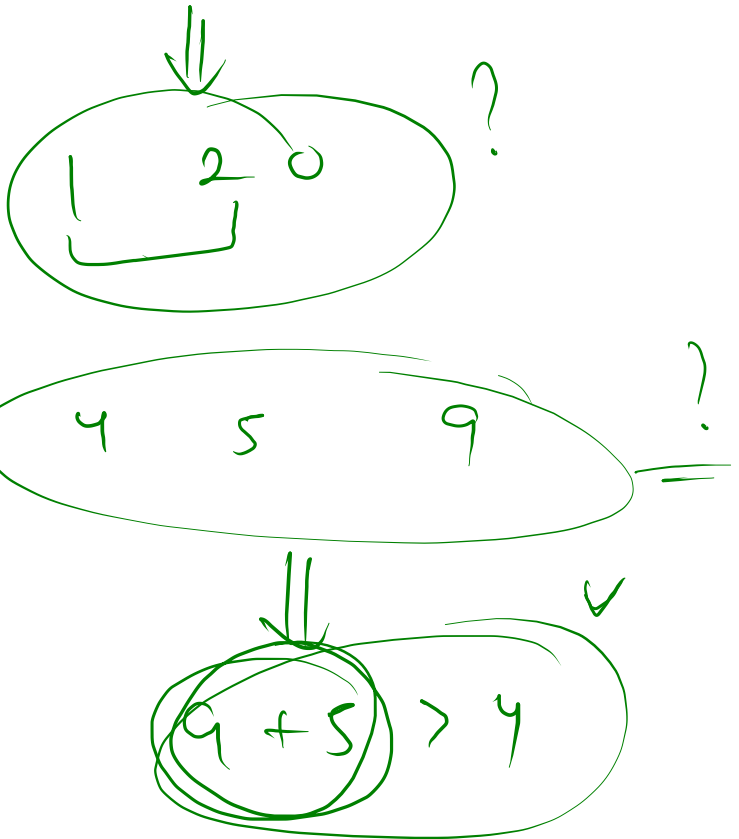

```
          7
        7 14
      7 14 21
    7 14 21 28
  7 14 21 28 35
7 14 21 28 35 42
```

7 14 21

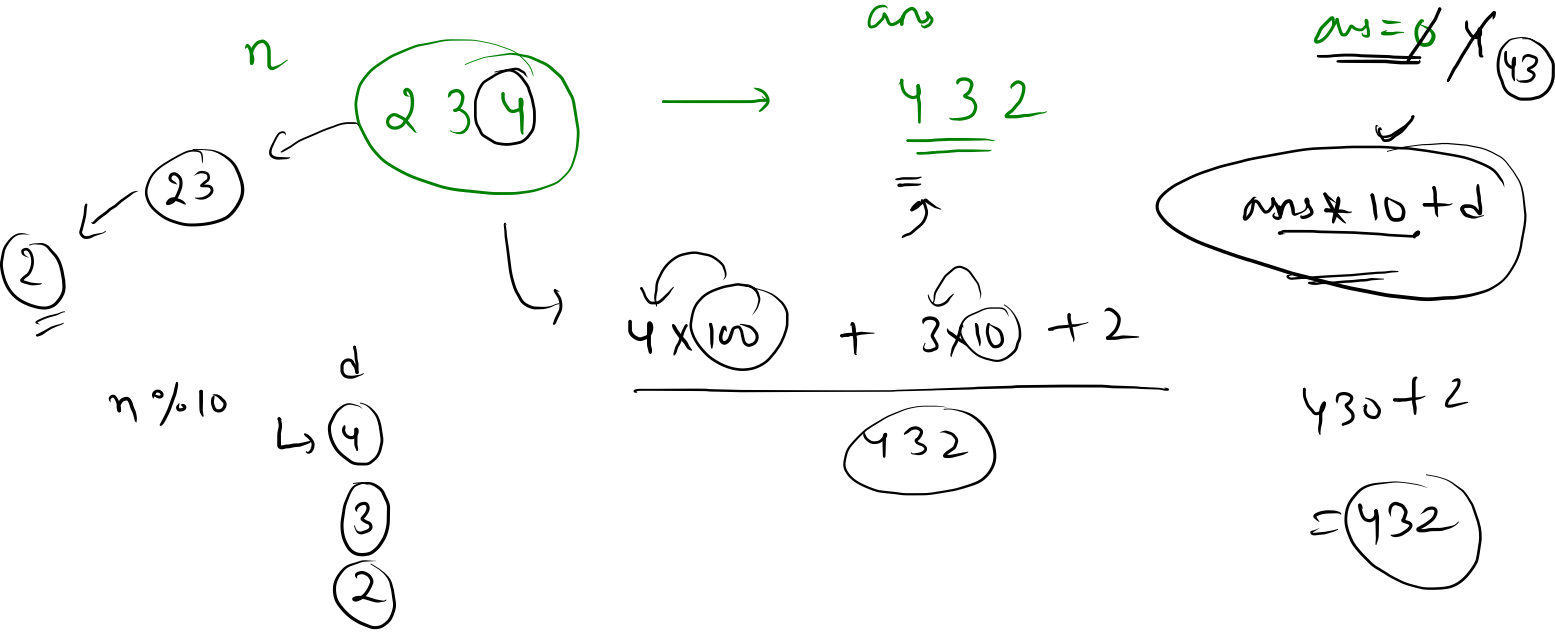
Triangle.

[0 1 2]

a+b > c || —



Reverse a number (Maths)



Maths

ans * 10 + d

123 → 321

3 × 100 (10 × 10)

+ 2 × 10 (10)

+ 1 ( )

d = 3

d = 2

d = 1

ans = ~~0~~ ~~3~~ ~~32~~ 321

3 × 100 (10 × 10)

123 → 321

d

3 x 100
2 x 10
1 x 1

300
20
1
321

count digits.

3

i = 1 → $10^{3-1} = 10^2 = 100$

3

for (2 times (n-1))

{

100

ans = 300 +20 +1 = 321 }

123

3 x 10

2 x 1

1 x 1

1

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static int reverse(int n){
        int ans = 0;
        while(n > 0){
            int d = n % 10;
            ans = ans * 10 + d;
            n = n/10;
        }
        return ans;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();//t=2

        for(int i = 1; i <= t; i++){
            int n = scn.nextInt();
            int ans = reverse(n);
            System.out.println(ans);
        }


    }
}
```

t=2

t times = 2 times.

2

5 7 9

4 6 3

1st.

=

n = 579 57 5 0

ans = 0 9 97 975

579>0

970

5

975

d = 9 5790

d = 7 570

d = 5 0>0

$n = 7$

$rows = n/2 + 1$

$= 4$

1st

$*$ $n-2$ $*$

Space $= n-2$

$\downarrow$

-2

-2

1 $*$

for (row = 1 $\longrightarrow$ $n/2+1$)

{

pre space.

star

space

Star

}

Permutation.

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

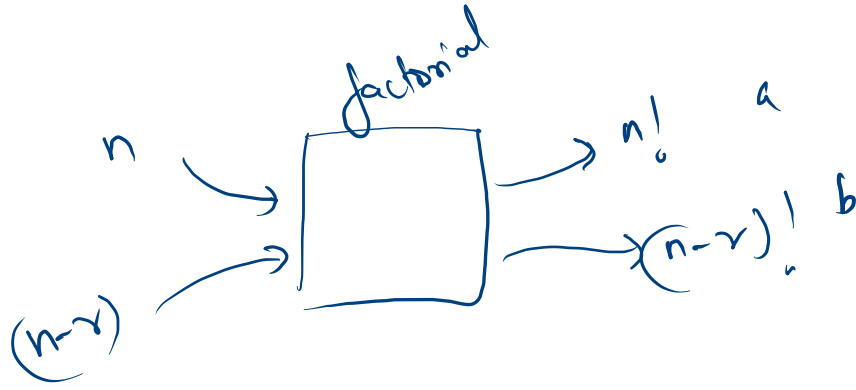
ans = a/b

Sample Input 0

```
n 5
r 2
```

Sample Output 0

```
20
```



```java
import java.io.*;
import java.util.*;

public class Solution {
    public static void reverse(int n){
        int ans = 0;
        while(n > 0){
            int d = n % 10;
            ans = ans * 10 + d;
            n = n/10;
        }
        System.out.println(ans);
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be n
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();

        for(int i=1; i<=t; i++){
            int n = scn.nextInt();
            reverse(n);
        }
    }
}
```