

## Revision

- Programming Language & Operators
- Variables
- Condition and logical operators
- ★ → if else condition (ladder)
- nested if else
- Switch case
- ★★ → Characters and Strings (typecasting & ascii value)
- ★ → for loops
- while and do while loop
- ✓ ★ → Patterns (Template)
- Functions (return statement)
- ✓ ★ → Digit Traversal & Number theory (%10 , /10)
- ★★★ → Arrays (Printing, updating, searching, upgradation)
- ✓ ★ → Brute force approach (Permutation & Combination)
- ★ → Time Complexity & Space complexity

Ques) Repeating & missing  
imp questions

- ↳ print inverted A
- ↳ diamond
- ↳ damrao

# → Interview flow

Friday :- OA (Hacker rank)  
Weekend :- Interview

no. of questions : 3

each of marks : 10 , total = 30 marks

passing % age : 60 %

→ must always go step by step

1) clearly understand and explain question

2) provide logic

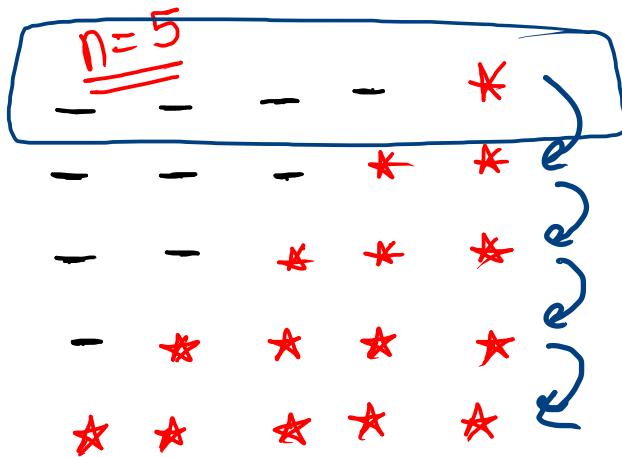
3) dry run → Time (max)



4) tell the time complexity

# Patterns

## (template)



int st = 1;  
int sp = n-1;  
for(int i=0; i < n; i++){  
 for(int j=0; j < sp; j++) {  
 System.out.print(" ");  
 }  
 for(int j=0; j < st; j++) {  
 System.out.print("\*");  
 }  
 System.out.println();  
 st++;  
}

*no. of rows*

# Hw\_Print Inverted triangle

$$\underline{n=7}$$

*	*	*	*	*	*	*
-	*	*	*	*	*	
-	-	*	*	*		
-	-	-	*			

$$\begin{array}{l} \vdots \\ n=9 \rightarrow 5 \\ n=7 \rightarrow 4 \\ n=5 \rightarrow 3 \\ \hline \hline n=3 \rightarrow 2 \\ n=1 \rightarrow 1 \end{array}$$

~~exp~~

$$(\text{rows} = \frac{n}{2} + 1)$$

int rows = (n/2) + 1 ;

int st = n ;

int sp = 0 ;

for(int i=0; i<rows; i++) {

    for(int j=0; j<sp; j++) {

        System.out.print(" "));

}

    for(int j=0; j<st; j++) {

        System.out.print("\* "));

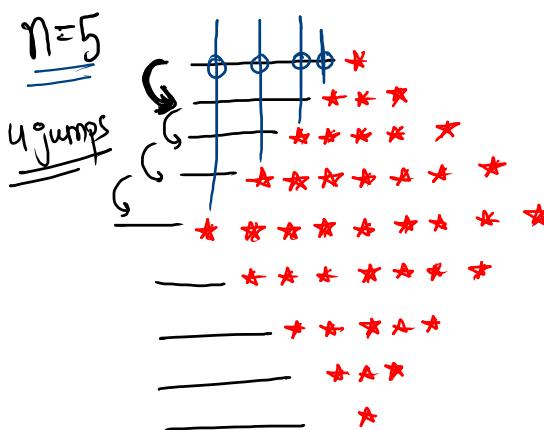
}

    sp++;

    st -= 2;

    System.out.println();

# GKSTR29\_Pattern\_12\_Diamond



$$\underline{\text{rows}} = 2n - 1 = 9/2 = 4$$

$i < \text{rows}$

0 1 2 3

int rows =  $2n - 1$ ;

int st = 1;

int sp =  $n - 1$ ;

for (int i = 0; i < rows; i++) {

    for (int j = 0; j < sp; j++) {

        System.out.print(" ");

    }

    for (int j = 0; j < st; j++) {

        System.out.print("\* ");

    }

    if ( $i < \text{rows}/2$ ) {

        st += 2;

        sp -= 2;

    } else {

        sp += 2;

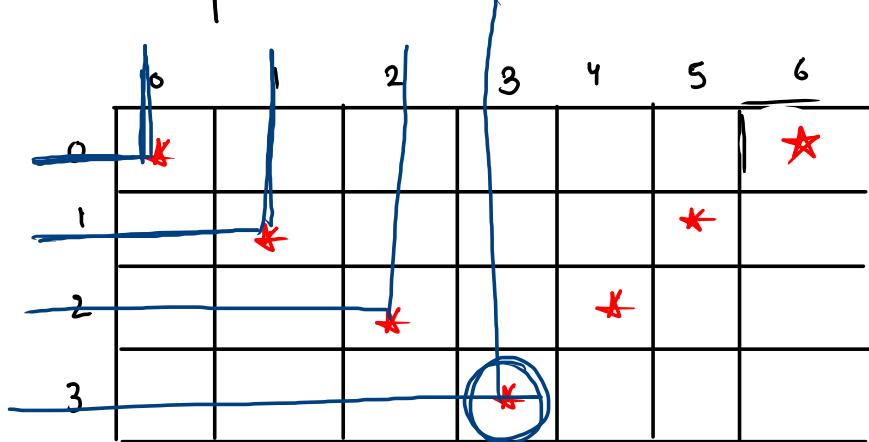
        st -= 2;

    }

    System.out.println();

$\Rightarrow \vee$  pattern

$$\underline{n=7}$$



for

$i$  —

    if ( $i == j$ ) || ( $i + j == n - 1$ )

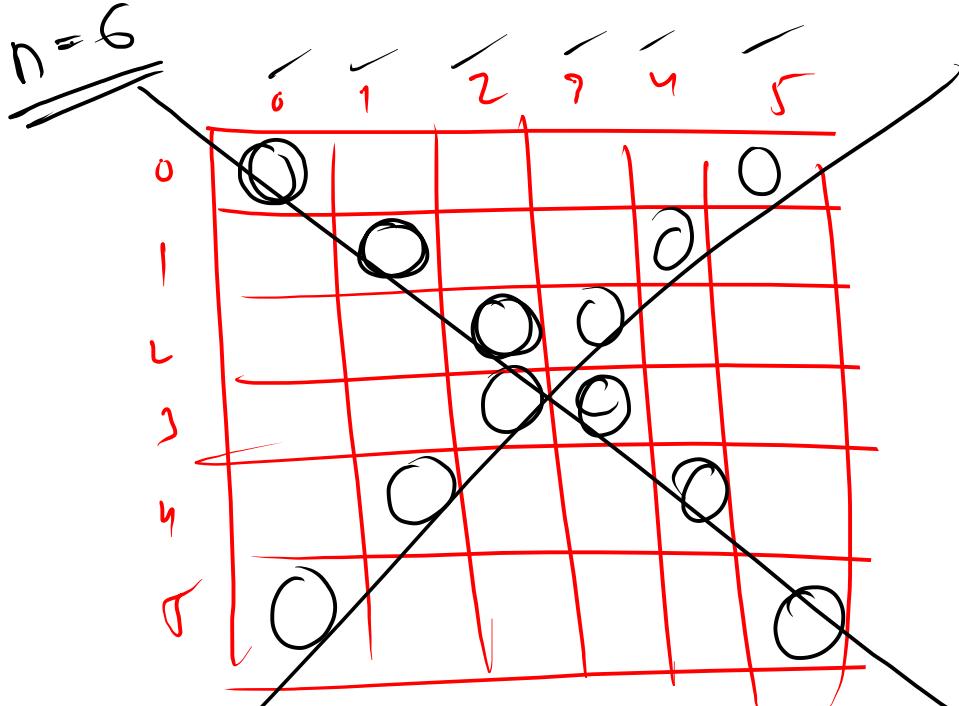
      —————  
      —————  
      \*

else

  —

add

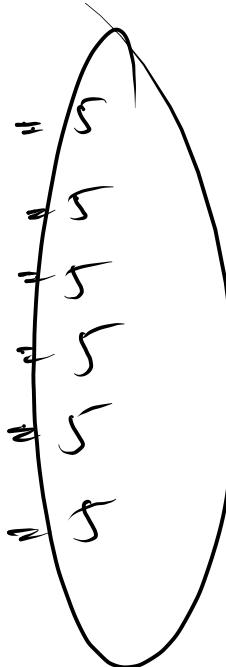
$$\left\{ \begin{array}{l} (3,3) = 6 \\ (2,4) = 6 \\ (1,5) = 6 \\ (0,6) = 6 \end{array} \right.$$



$$i+j = n-1$$

$$i == j$$

$$\begin{array}{lcl}
 5,0 & = 5 \\
 4,1 & = 5 \\
 3,2 & = 5 \\
 2,3 & = 5 \\
 1,4 & = 5 \\
 0,5 & = 5
 \end{array}$$



$\Rightarrow$  digit traversal

Reverse n digit no.

$$n = 4$$

$$\underline{\underline{num = 0}}$$

$$(num = num * 10 + val)$$

$$val \rightarrow 3$$

$$, \quad num = 0 * 10 + 3 = 3$$

$$val \rightarrow 7$$

$$, \quad num = 3 * 10 + 7 = 37$$

$$val \rightarrow 0$$

$$, \quad num = 37 * 10 + 0 = 370$$

$$val \rightarrow 1$$

$$, \quad num = 370 * 10 + 1 = 3701$$

$$\text{num} = \underline{\underline{3701}}$$

$$, \quad \text{ans} = \underline{\underline{0}}$$

$$\text{num} > 0$$

$$\text{rem} = \underline{\underline{\text{num} \% 10}}, \quad \text{ans} = \underline{\underline{\text{ans} * 10 + rem}}, \quad \text{num} / = \underline{\underline{10}}$$

$$\text{rem} = \underline{\underline{3701 \% 10}}, \quad \text{ans} = \underline{\underline{0 * 10 + 1}}, \quad \text{num} = \underline{\underline{370}} \\ = 1 \qquad \qquad \qquad = 1$$

---

$$\text{rem} = \underline{\underline{370 \% 10}}, \quad \text{ans} = \underline{\underline{1 * 10 + 0}}, \quad \text{num} = \underline{\underline{37}} \\ = 0 \qquad \qquad \qquad = 10$$

---

$$\text{rem} = \underline{\underline{37 \% 10}}, \quad \text{ans} = \underline{\underline{10 * 10 + 7}}, \quad \text{num} = \underline{\underline{3}} \\ = 7 \qquad \qquad \qquad = 107$$

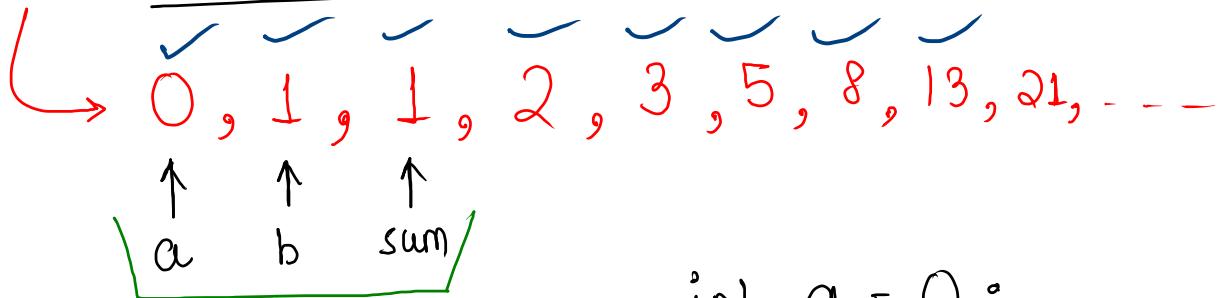
---

$$\text{rem} = \underline{\underline{3}}$$

$$\boxed{\text{ans} = \underline{\underline{1073}}}$$

$$\text{num} = \underline{\underline{0}} \times$$

$\Rightarrow$  Fibonacci series (sum of last 2 digits)



$\frac{n=8}{}$

int a = 0 ;

int b = 1 ;

int sum = 0 ;

for (int i=0; i<n; i++) {

    print a;

    sum = a+b;

    a = b

    b = sum

# → functions

fun(); → call

int a = 5;  
int[] arr = {}

public static ~~void~~ int[] fun() { → declare  
by }  
return type

void → nothing

int, boolean,  
char, String, int[]

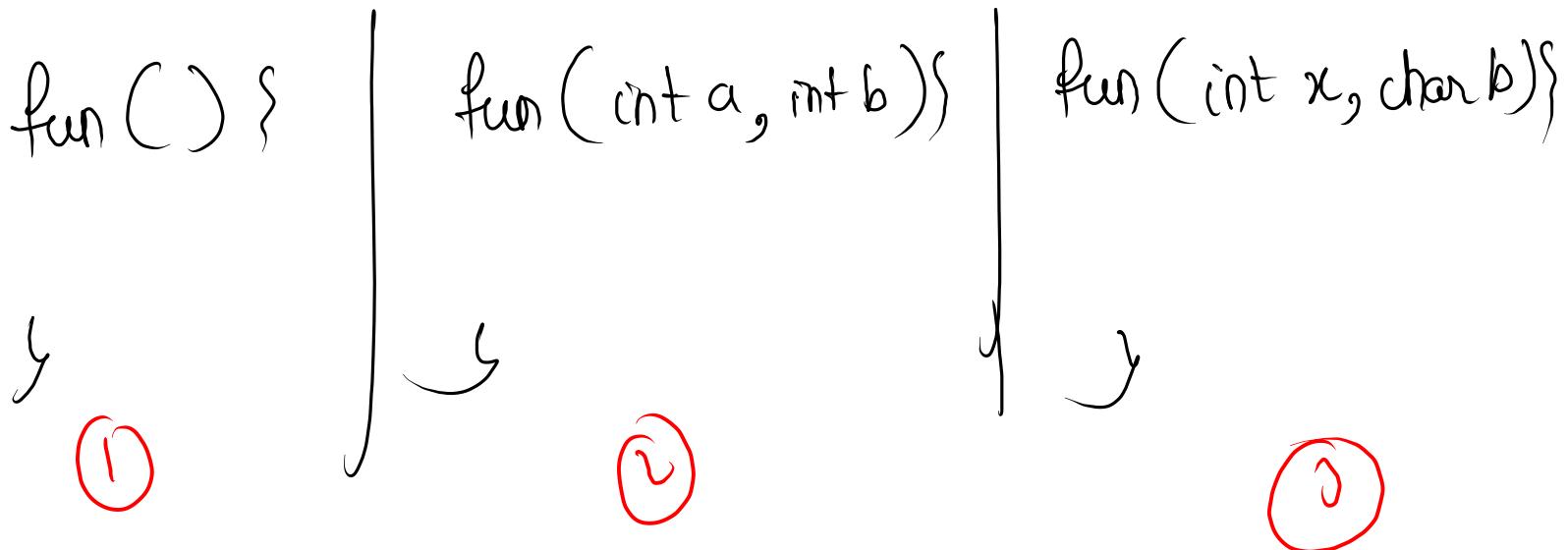
## fun calling

fun() ←

fun(s, 'a'); ←

fun(3, 4); ←

## Overloading



# HW\_Repeating and Missing element

P wtr

$$arr = [5, 3, 5, 1, 8, 2, 4, 6, 9]$$

sum of first  $n$  no.  $\Rightarrow \frac{n(n+1)}{2}$

miss = 7  
rep = 5 ←

$$\Rightarrow \frac{9 * (9+1)}{2} = \underline{\underline{45}} \quad (\text{expected sum})$$

$$= 43 \quad (\text{actual sum})$$

$$= 5 \quad (\text{repeated})$$

missing no. = expected sum - actual sum + MRP

$$= 45 - 43 + 5$$

$$= 7$$