** Operators

① Arithmetic operators.

→ Binary operators (+ - * / %)

→ Unary operators (++, --)

Incrementers

Decrementers

Pre-Increment (++a)

~~Istanst~~

```
int a=10;
int b;
b = ++a
```

b = 11

Post- Increment (a++)

```
int a=10;
int b;
b = a++;
```

b = 10

② Relational operators.

Return a boolean value.

6 operators

==	Compare two value.
!=	Not equal to
>	-
<	-
>=	-
<=	-

③ Logical operators.

&&	AND gives us true if both operand true.
||	OR gives us true any one.
!	NOT gives opposite logical value of the operand.

④ Bitwise operators.

& AND

```
  0101
& 0110
------
  0100
```

| OR

```
  0101
| 0110
------
  0111
```

^ XOR operator

```
  0101
  0110
------
  0011
```

~ ones Complenet.

```
~ 0101
------
  1010
```

<< left shift

4 << 1 → shift (0100) 1 pos.
= 1000

>> Right Shift.

4 >> 1
(0100)
= (0 010)

a << n → a * 2^n

a >> n → a / 2^n

(5) Assignment operators.

=

+=

-=

*=

/=

Sum = Sum + Hello;

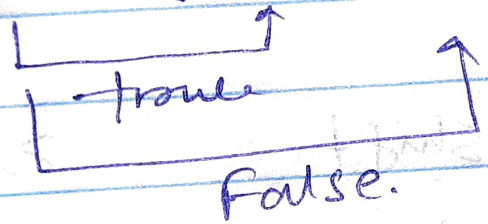
Sum += Hello;

⑥ Miscellaneous.

Sizeof() Return size of variable.

✓ ~~Condition~~ Condition
Ternary operator.

Condition ? x : y



& (Return the address of a var)

* (pointer to a variable)

* a = (2, 3, 4)
Here assign '4' into a.

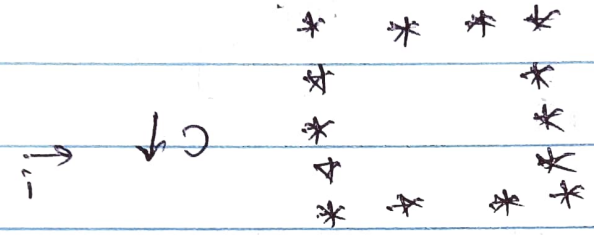
PATTERN Question

① Rectangle pattern.

(two for loops)

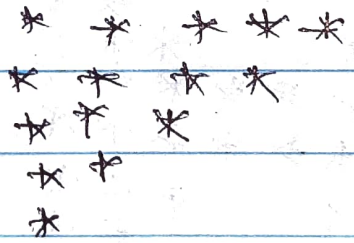
(easy)

② Hollow Rectangle pattern.

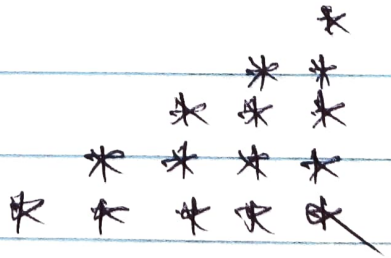


```
for i=1 to n
    J=1 to n
    Print Star
```

③ Inverted Half pyramid.



④ Half pyramid after 180° Rotn.



(5) Half pyramid using NO

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

(Increase i value)

(6) Floyd's Triangle.

```
1
2 3
4 5 6
7 8 9 10
```

Normal. easy. (extra variable)

(7) Butterfly pattern.