Methods In Java Contd...

I/p = 123   O/p = 321

* Reverse an Integer :→

```
int reverse_integer (int n)
{

}
```

int digit = n%10
digit = 3
n = n/10 = 123/10
= 12
digit = n % 10
= 12 % 10 = (2)
n = n/10 = 12/10 = 1
digit = n % 10 = 1 % 10 = 1
n = n/10 = 1/10 = 0 → stop

123 = 1×100 + 2×10 + 3
int ans = 0;
ans = ans×10 + digit;
= 0×10 + 3 = 3;
= 3×10 + 2 = 32;
= 32×10 + 1 = 321
(TCs)

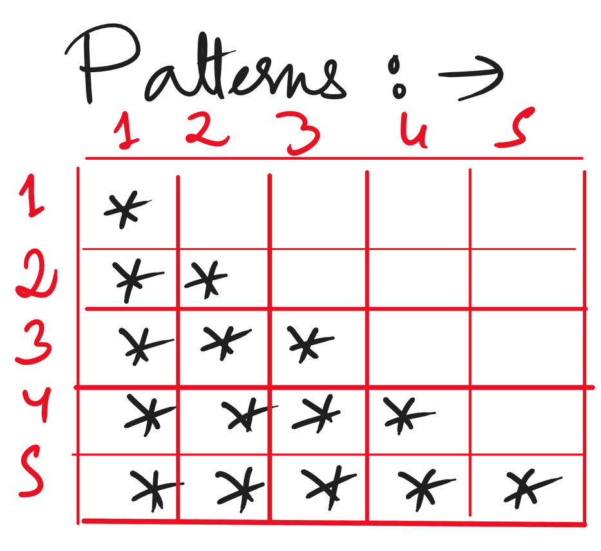
n = 452
n = n/10
= 452/10
= 45
n = n/10
= 45/10
= 4
n = n/10
= 4/10
= 0 stop

ans = 0
ans = ans×10 + digit
ans = 0×10 + 2
= 2
ans = ans×10 + digit
= 2×10 + 5
= 25
ans = ans×10 + d
= 25×10 + 4
= 254

digit = n % 10
= 452 % 10
= 2
digit = n % 10
= 45 % 10
= 5
digit = n % 10
= 4 % 10
= 4

Patterns :→

n = (5)



Row = 1 to n
Col = 1 to Row Number

```
for ( int r=1; r<=n; r++) {
    for (int c=1; c<=r; c++){
        sout ( " * " );
    }
}
```

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

Identity Matrix :→

(n = 4)

	1	2	3	4
1	(1)	0	0	0
2	0	(1)	0	0
3	0	0	(1)	0
4	0	0	0	(1)

r c
[1,1]
[2,2]
[3,3]
[4,4]

```
for (int r=1; r<=n; r++){
    for (int c=1; c<=n; c++)
    {
        if (r==c)
            print (1);
        else
            print (0);
    }
```

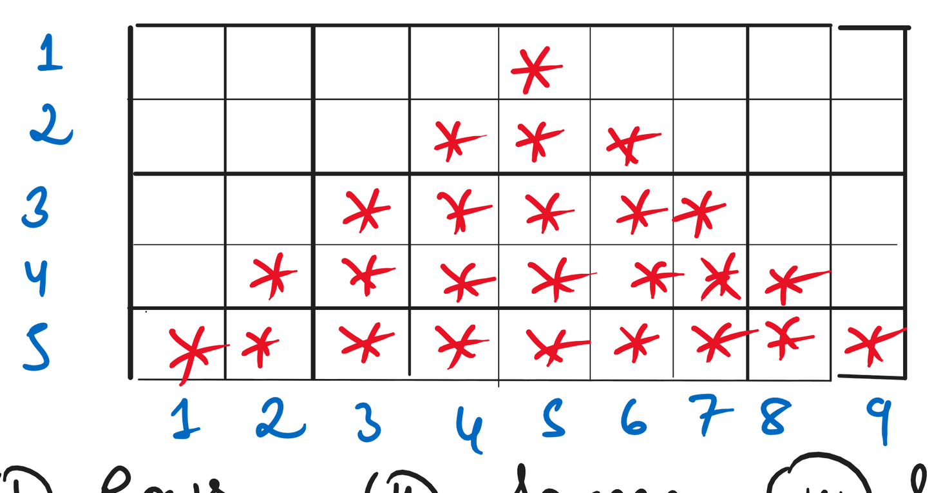
Hollow Square Pattern :

n = 4

```
* * * *
*     *
*     *
* * * *
```

→ Print stars (*) when we reach 1st & last row & colm

[(n=5)]   (Pyramid) (Dynamic Patterns)

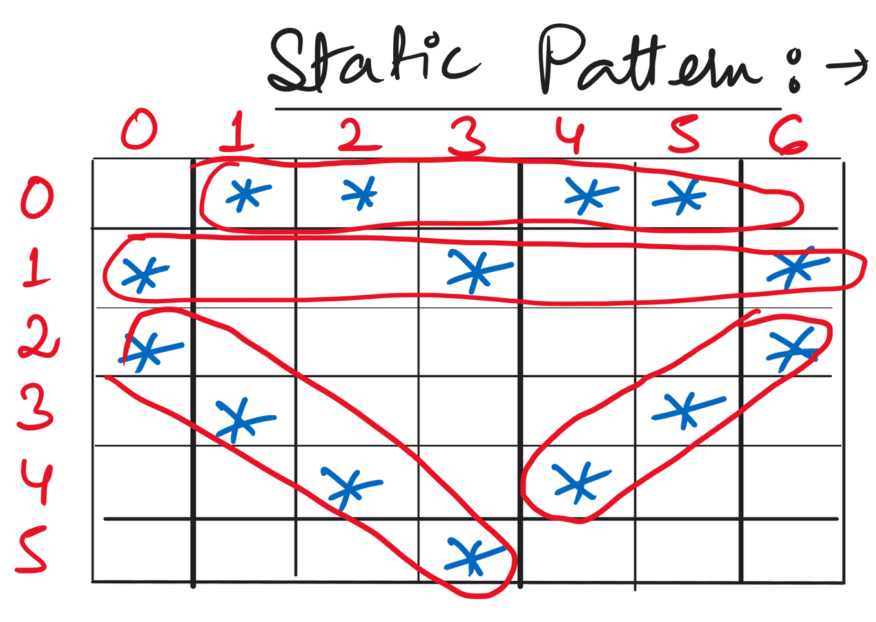


(i) Rows: i = 1, 2, 3, 4, 5
(ii) Spaces: n−i → 4, 3, 2, 1, 0
(iii) Stars: 2*i−1 → 1, 3, 5, 7, 9

```
for (int i=1; i<=n; i++)
{
    // spaces
    for (int j=1; j<=(n-i); j++)
        print (" ");
    // stars
    for (int k=1; k<=(2*i-1); k++)
        print (" * ");
}
```

Static Pattern :→

[6 Rows 7 Cols] Will not change



Rule 1: Break into smaller similar parts.

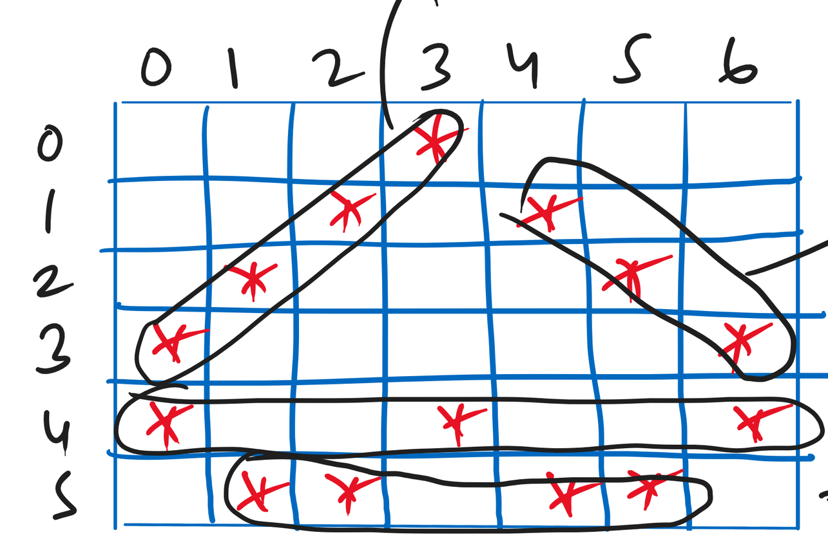
Time		Pkg
4 m	TCS NQT	5-6
5 m	TCS Ninja	3-4
3 m	TCS Code Vita	6
4 m	Accenture	4-6
5 m	Capgemini	3-4
3 m	Oracle	8
(mins)		LPA

P1: r==0   c = 1,2,4,5   c%3!=0

P2: r==1   c = 0,3,6   c%3==0
print (" * ")
else (" ")

P3: r−c ==2
2 − 0
3 − 1
4 − 2
5 − 3

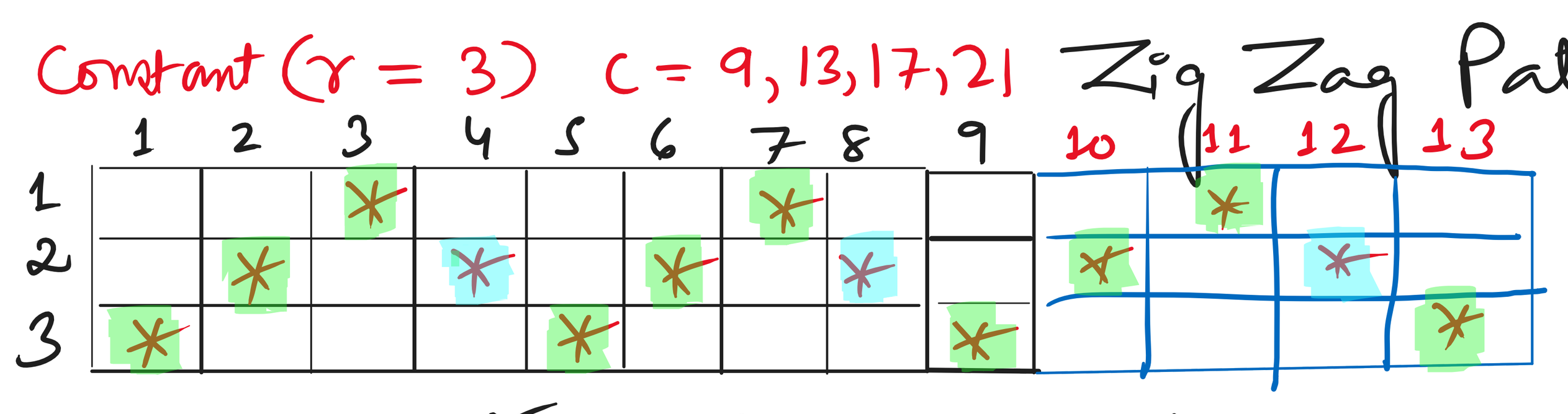
P4: r+c ==8
2 + 6
3 + 5
4 + 4



r+c == 3
(r−c) (−3) or (c−r) (3)

⇒ r==4 & c%3 == 0
⇒ r==5 & c%3 != 0

Constant (r = 3) c = 9,13,17,21 Zig Zag Pattern



Bosch, Accenture, LG Soft, Oracle } 6-8 LPA

Rule 2:→ If time is less, don't follow Rule 1.   2 mins (Magic No.)

R1	3	7	11	⇒	c%4==3
R2	Even	No			c%2 ==0
R3	1	5	9	13	c%4 ==1

{ [*] (r+c)%4 == 0

[*] r==2 && c%4==0 }

When to use static :→

When we write inside main() :→