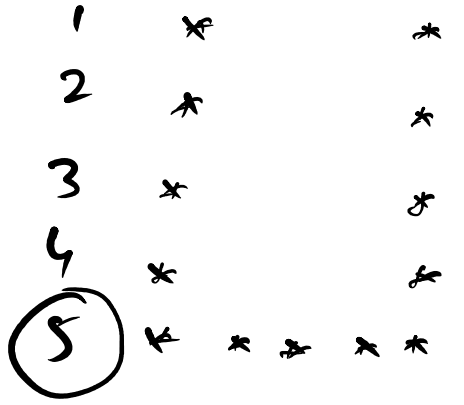
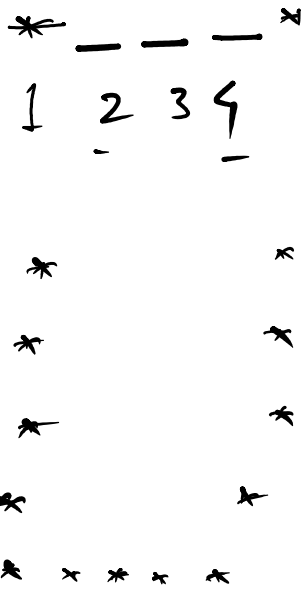
Pattern 8 - Print a hollow square without top

n = 5 (5 × 5) -> Square.



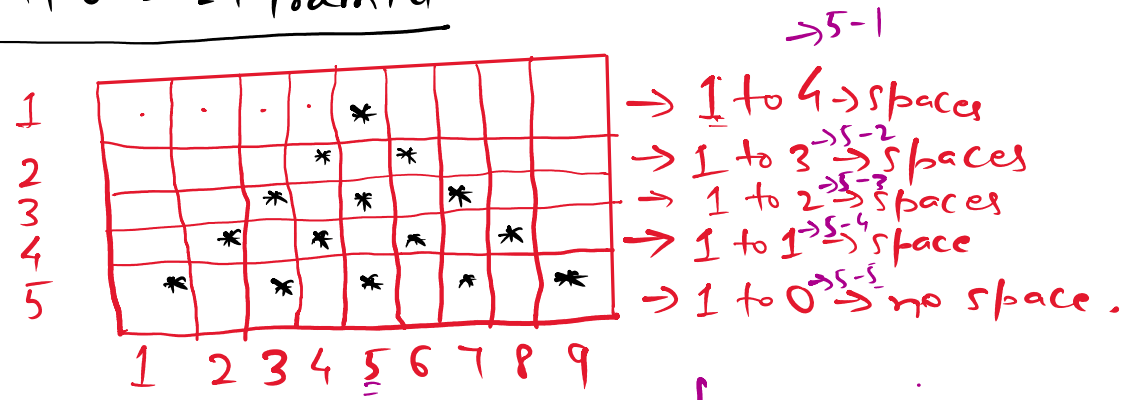
```
for(int i=1;i<=5;i++)
for(int j=1;j<=5;j++)
```

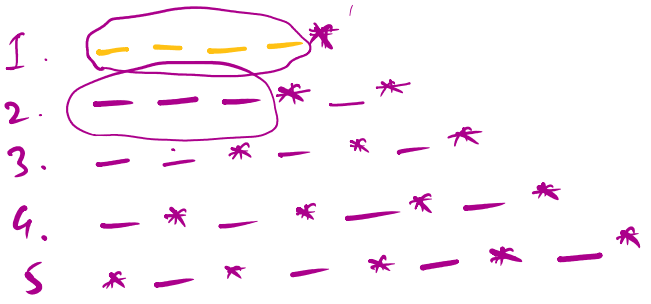
```
for(int i=1;i<=n;i++){
    if(i==n){ -> Last line.
    for(int j=1;j<=n;j++){
        S.o.p("*");
    }
    }else{
        S.o.p("*");
    for(int j=2;j<=n-1;j++){
        S.o.p(" ");
    }
    S.o.p("*");
```



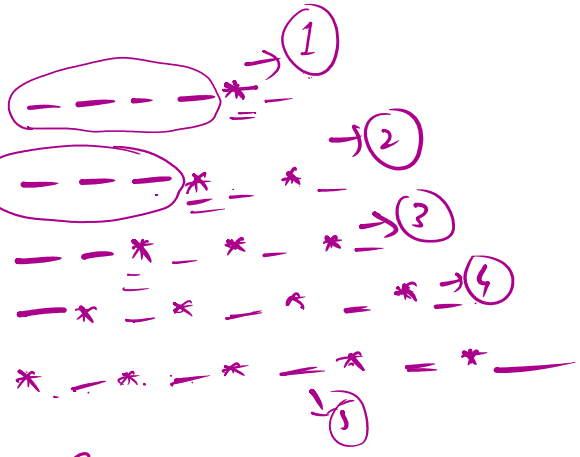
```
        s.o.p( * );
    }
    s.o.println();
}
```

Pattern_7_Pyramid

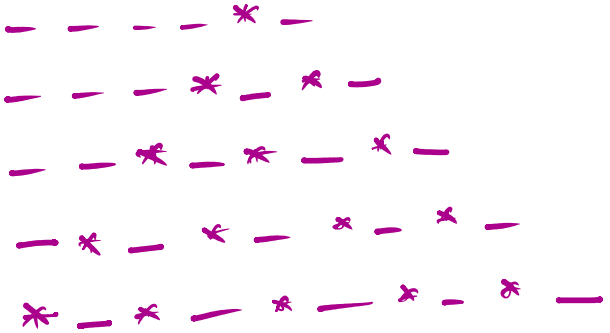





```
for(int i=1;i<=n;i++){
    for(int j=1;j<=n-i;j++){
        S.o.p(" ");
    }
    for(int K=1;K<=i;K++){
        S.o.p("* ");
    }
}
```



Dry Run



Diamond Pattern

1 - 1 → 1×2-1 = 1

2 - 3 → 2×2-1 = 3

3 - 5 → 3×2-1 = 5

4 - 7 → 4×2-1 = 7

5 - 9 → 5×2-1 = 9.

k → 1 to i*2-1

for(int i: 1 to 5){

}

for(int i: 1 to n-1)

for(j: 1 to i)

→ n: 4

1 → 7 → 2×(n-i)+1 → 2×(4-1)+1 = 2×3+1 = 7

2 → 5 → 2×(n-i)+1 → 2×(4-2)+1 = 2×2+1 = 5

3 → 3 → 2×(n-i)+1 → 2×(4-3)+1 → 2×1+1 = 3

4 → 1 → 2×(4-4)+1 = 2×0+1 = 1

Functions

It is used when we have some lines code to reused again and again.

function declaration :- We specify the return type, name and parameters.

return type -> Which data type function is going to return

```
return_type function_name(parameters) {
// statement
}
```

a = 2
b = 3 , a+b = 2+3 = 5 -> integer.

int -> return type

```
int sum( ) {
```

-> not returning anything

```
void sum( )
```

```
string function1( )
```

Ex:-

```
void function1(  ){
// S.o.Pln("Hello");
//S.o.Pln("Hi Hello").
}
```