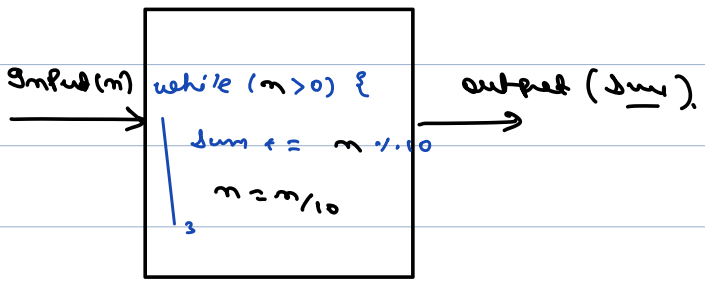
← functions →

```
main() {
    . . . . .
    while (a1>0) {
        sum1+= a1 %10
        a1 = a1/10
    }
    Print (sum1);
    . . . . . .
    while (a2>0) {
        sum2+= a2 %10
        a2 = a2/10
    }
    Print (sum2);
    . . . . . .
    while (a3>0) {
        sum3+= a3 %10
        a3 = a3/10
    }
    Print (sum3);
}
```

Problems

- Redundancy
- Readability.
- Maintainability.



int → syntax

sumofdigit → functionName

int → inputType

n → input

```
anyType functionName (inputType input) {
    // write logic;
    return ans;   → sum
}
```

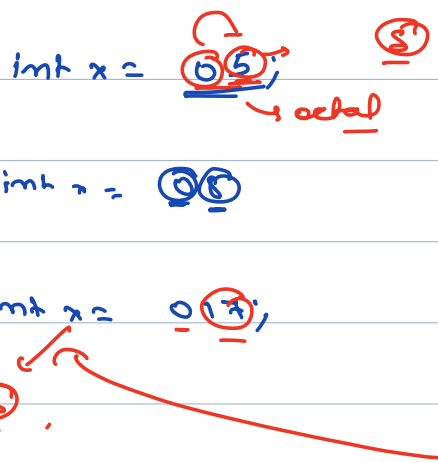
// sum of 2 integers.

function / method.

```
int int2sum (int a, int b) {
    int sum = a+b;
    return sum;
}
```

```java
public static int sum3(int a, int b){
  System.out.println(a+b);
  return 0;
}

public static void main(String[] args) {
    int a = 10;
    int b = 15;
    // int ans = int2Sum(a,b);
    // System.out.println(ans);
    // System.out.println(int2Sum(a,b));
    // System.out.println(sum3(a,b));
    sum3(a,b);
}
```



n = 5

```
********** // 0 spaces

****__**** // 2 spaces

***____*** // 4 spaces

**______** // 6 spaces

*________* // 8 spaces
```

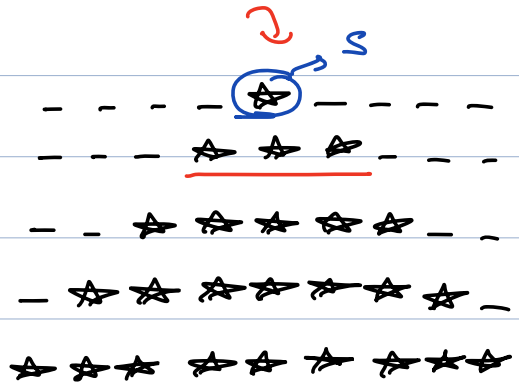
row	st1	sl	st2
1	5	0	5
2	4	2	4
3	3	4	3
4	2	6	2
5	1	8	1

st1 = n+1 − row

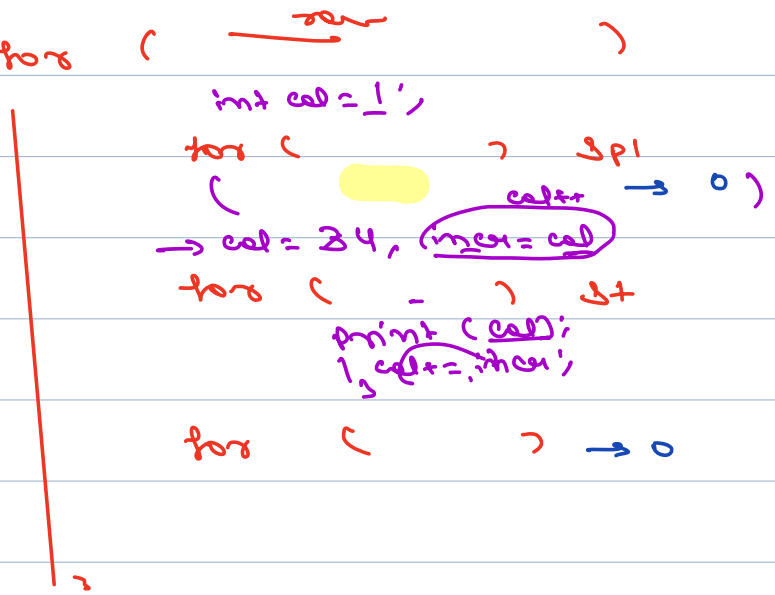
2 * row − 2

n = 5

1	0	0	0	0	5	0	0	0	0
2	0	0	0	4	8	12	0	0	0
	0	0	3	6	9	12	15	0	0
	0	2	4	6	8	10	12	14	0
	1	2	3	4	5	6	7	8	9



2 B



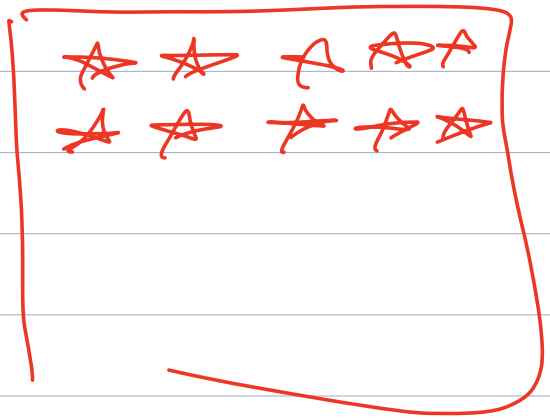
1 2 3 4 5

11 12 13 14 15

21 22 23 24 25

16 17 18 19 20

6 7 8 9 10



```
startNo = 1;
for (i = 1; i <= n; i++) {
    int colNo = startNo;
    for (j = 1; j <= n; j++) {
        print (colNo);
        colNo++;
    }
    startNo += 10;
}
```

startNo * 11

3

row

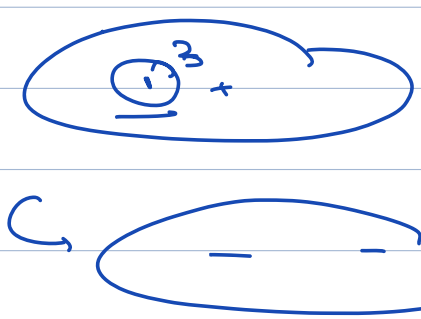
1	1	2	3	4	5
2	11	12	13	14	15
3	21	22	23	24	25
	16	17	18	19	20
	6	7	8	9	10

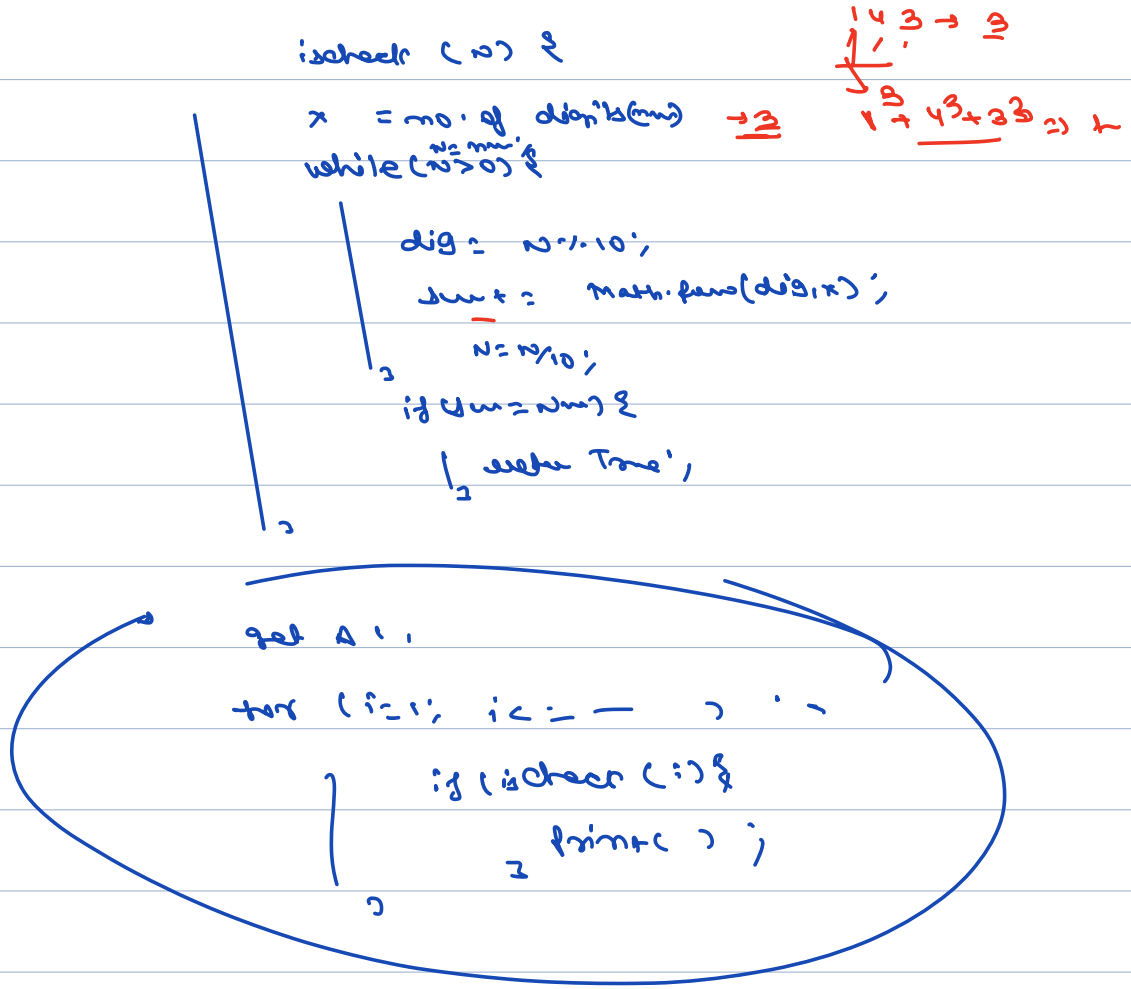
1 2 3 4 5 6

7 8 9 10 11 12

153

$1^3 + 5^3 + 3^3 \Rightarrow 153$.





https://www.scaler.com/topics/java/online-java-compiler/?snippet_slug=d2ca34668275c1ab79cd