Abbreviation using backtracking:

string ⇒ pep

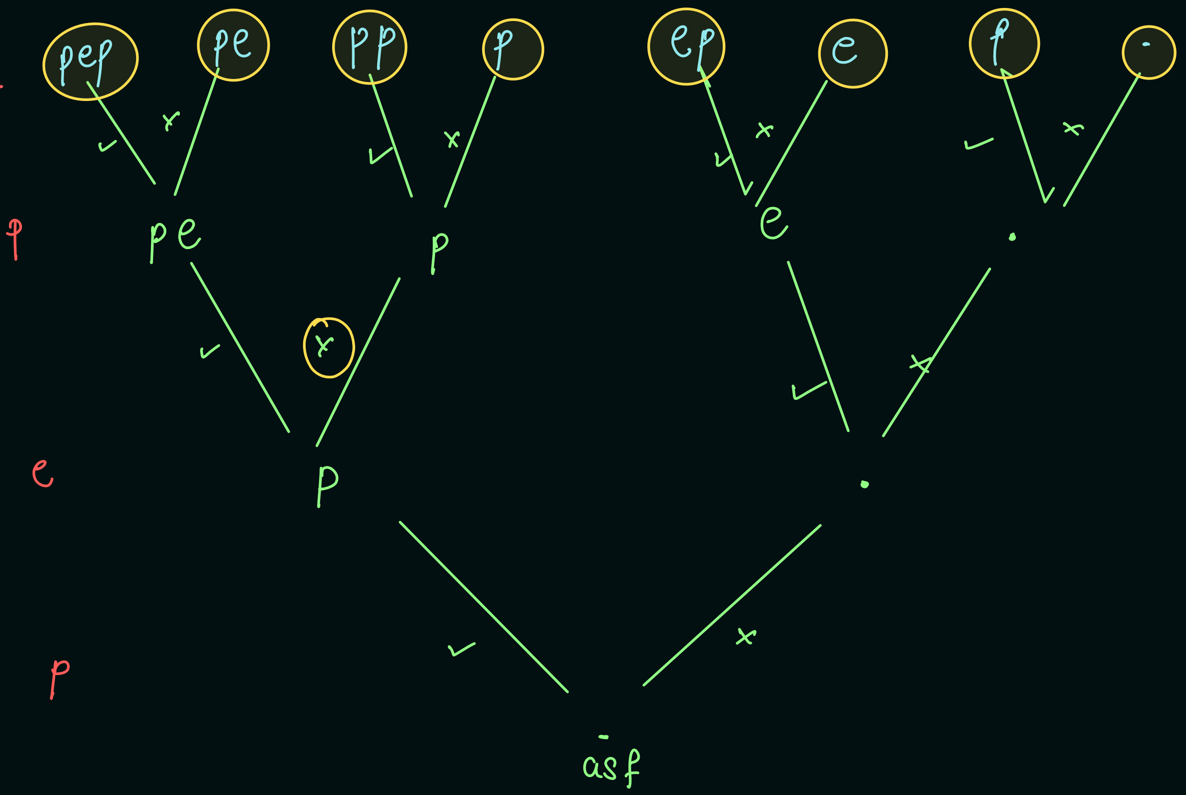
Base Idea of problem is → Subsequence.

level → character

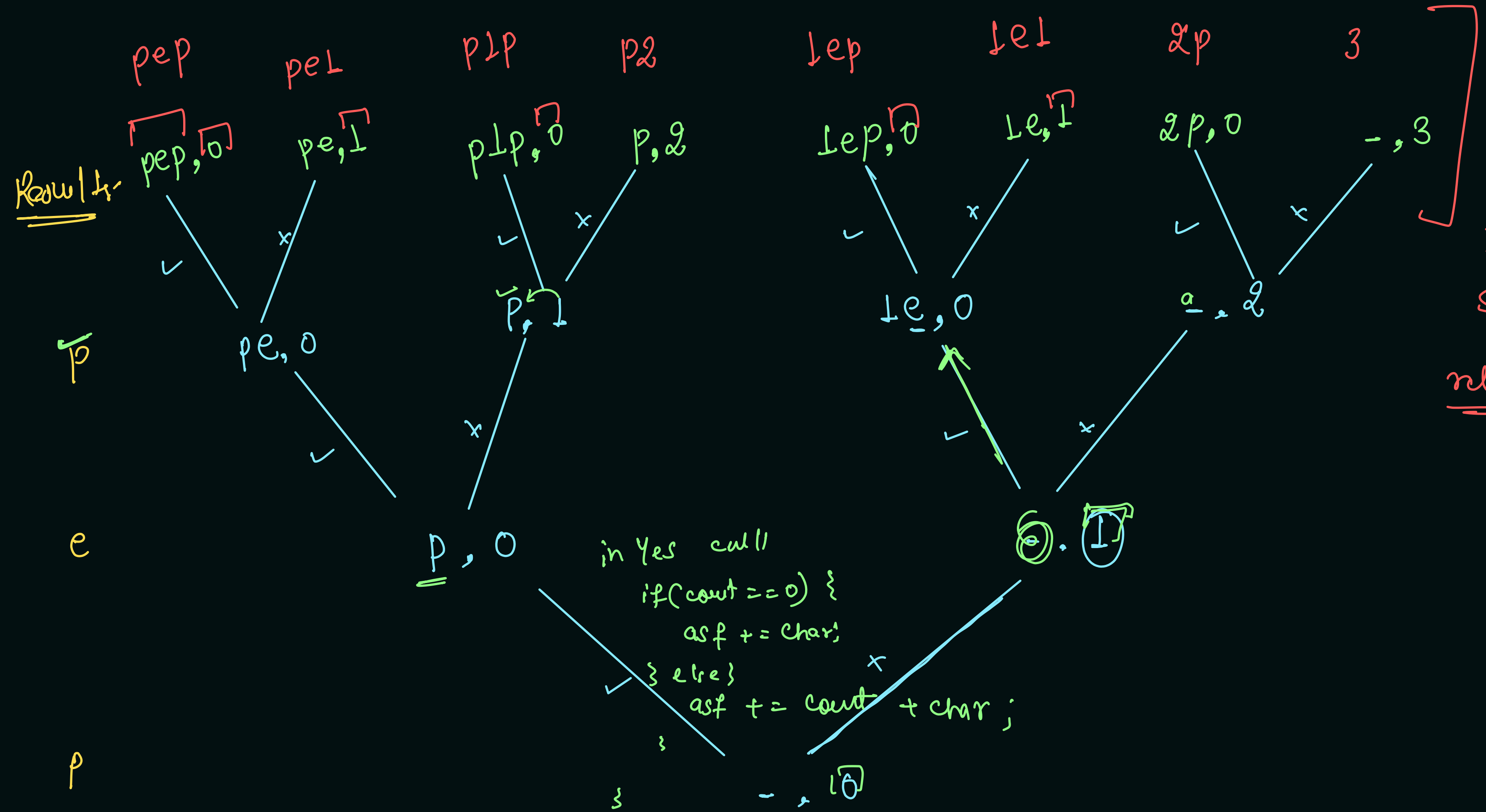
option → Yes and No of characte

0 0 0	pep	pep
0 0 1	pe1	pe
0 1 0	p1p	pp
0 1 1	p2	p
1 0 0	1ep	ep
1 0 1	1e1	e
1 1 0	2p	p
1 1 1	3	.

Results



How we can print numerical value.



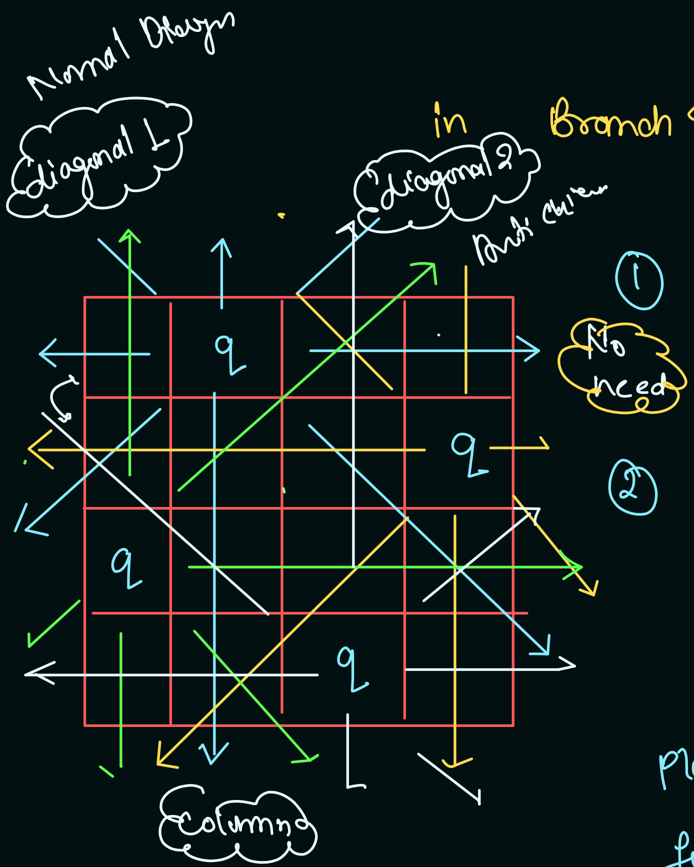
```
in Yes call
    if(count == 0) {
        asf += char;
    } else {
        asf += count + char;
    }
}
```

answer so far,   count

```
if(count != 0){
    asf += count;
}
Syso(asf);
return;
```

N - Queen Branch and Bound:

Complexity of " IsSafetoplace Queen" in previous question ? nearly = 4n ≡ O(n)



in Branch & Bound, we will reduce complexity from O(n) to O(1)

① In a row, we have to place Exactly one queen for sure. we can't place more than one queen in a row

② In next row, we have to check if position is safe or not.

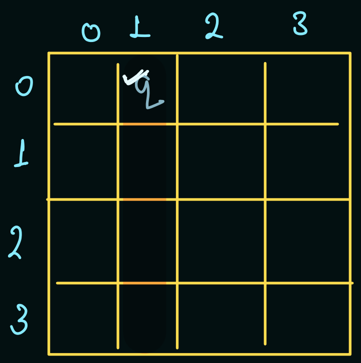
Rows are at level, & Row will change at every level, place queen 'q' and stop column & diagonals for future queen, this whole logic is called "Branch & bound" method.

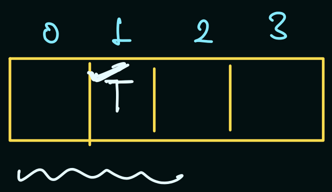
How to check isSafe in O(1)

diagonal 2 → normal diagonal [from top right to bottom left]

diagonal → Reverse diagonal [from top-left to bottom-right]

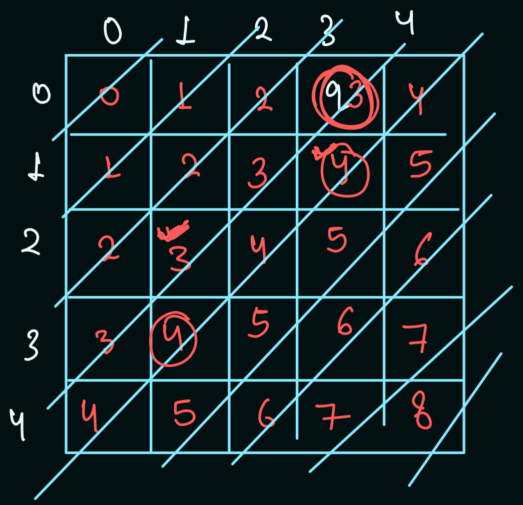
① How to Ensure Safety of Column;





booleam [] col = new booleam [column.length];

② How to Ensure Safety of Normal diagonal [from top-right to bottom left]



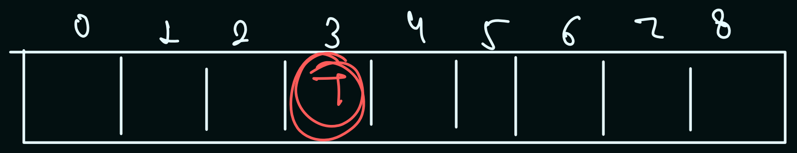
	0	1	2	3	4
0	0	1	2	Q3	4
1	1	2	3	4	5
2	2	3	4	5	6
3	3	4	5	6	7
4	4	5	6	7	8

→ direction

no. of diagonal = row + col

lenth of diagonal array to check suffety is

2 * length - 1 = 2× 5 - 1 = 9



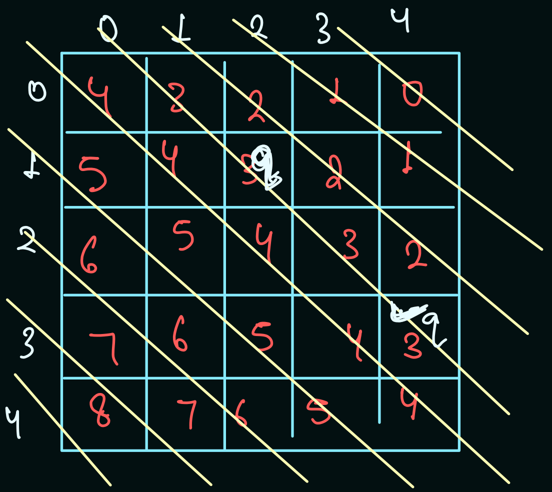
③ How to Ensure Safety in Reverse diagonal [from top-left to Bottom - Right]

	0	1	2	3	4
0	0	-1	-2	-3	-4
1	1	0	-1	-2	-3
2	2	1	0	-1	-2
3	3	2	1	0	-1
4	4	3	2	1	0

row - col → direction of diagonal

How to find proper index

↳ (row - col) + (length - 1)



	0	1	2	3	4
0	4	3	2	1	0
1	5	4	3	2	1
2	6	5	4	3	2
3	7	6	5	4	3
4	8	7	6	5	4

length of diagonal = 2 * length - 1 = 2 × 5 - 1 = 9

0	1	2	3	4	5	6	7	8
			ⓣ					

(1, 2)

Rev. diag ind = (row - col) + (length - 1)

= (1 - 2) + (5 - 1)

= ③

to place a queen at i,j index

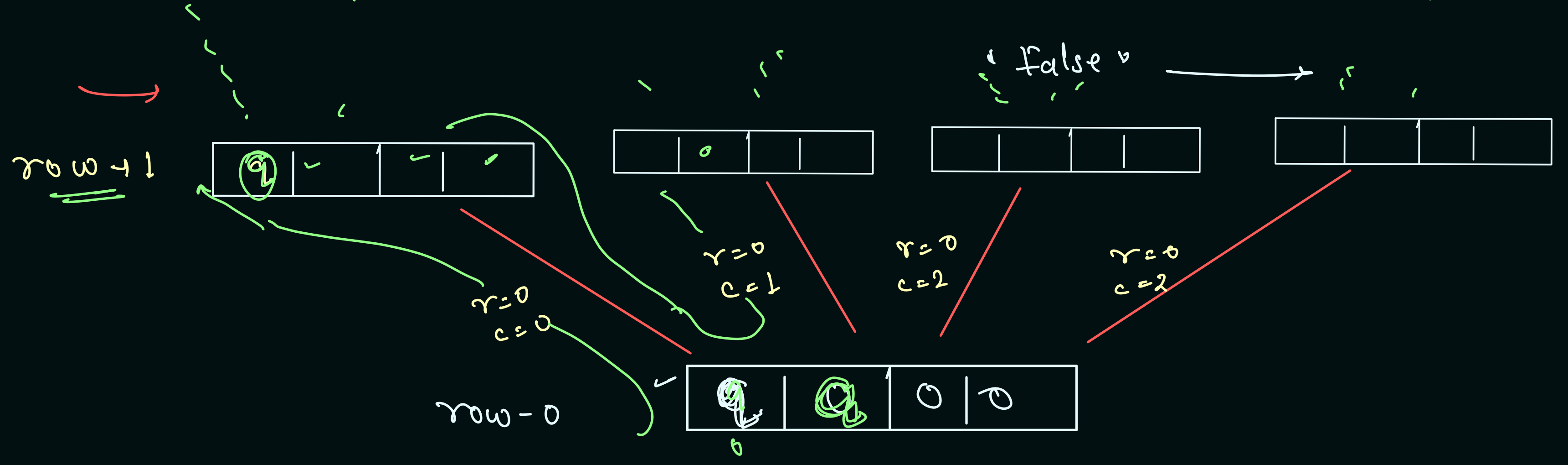
↳ pass three check →
① check of colum
② check of diagonal
③ check of rev. diagonal

→ if check is pass

↳ then make a mark on
① colum → col[j] = true
② Diagonal → dia1[i+j] = true
③ Reverse diag
↳ dia2[i-j+ length-1] = true

make call

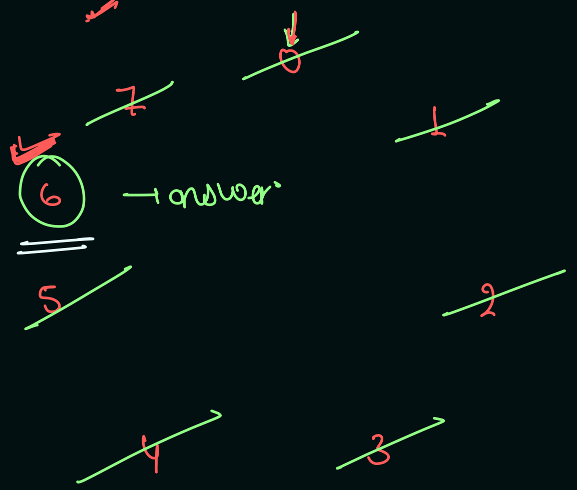
→ unmark → ① colum ② diagonal & ③ Rever. diagonal



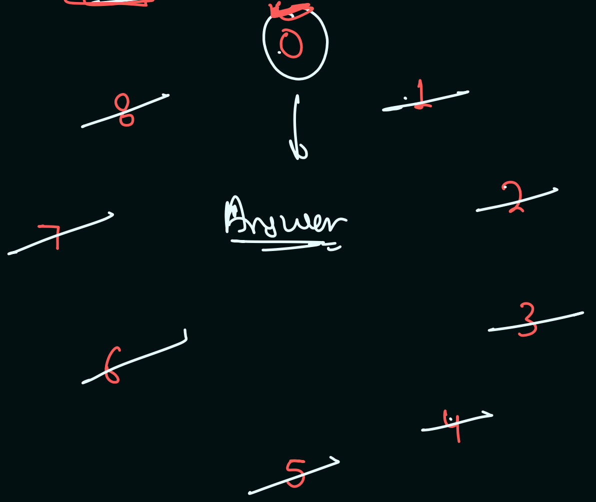
Josephus Problem:

n= no. of people in circle , k→ number.

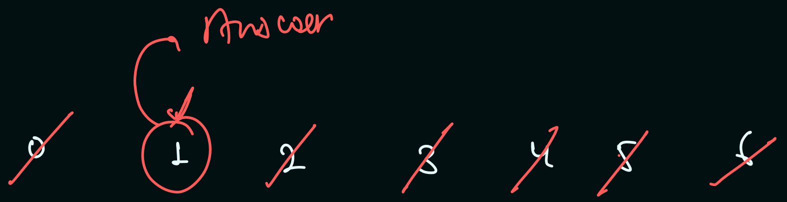
n=8, k=3



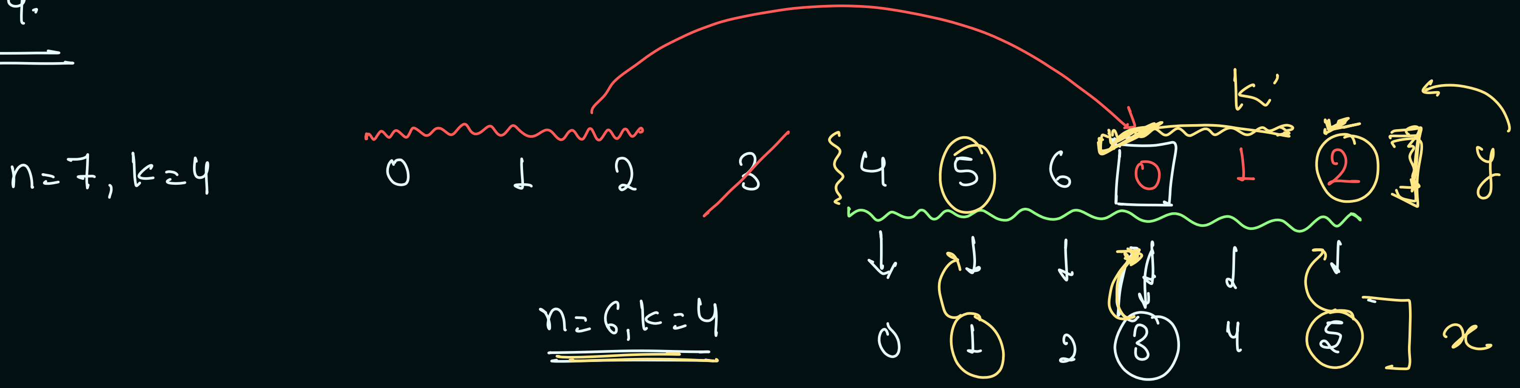
n=9 , k=4



n=7, k=4



n=7, k=4.



$y = (x + k) \% n$ in current level ;

$x=3$

$y = (3+4) \% 7 = 7 \% 7 = 0$ $\quad$ $y = (1+4) \% 7$

$= 5 \% 7 = 5$

$y = (5+4) \% 7 = 9 \% 7 = 2$

Expectation → (n, k) → find remain perrson.

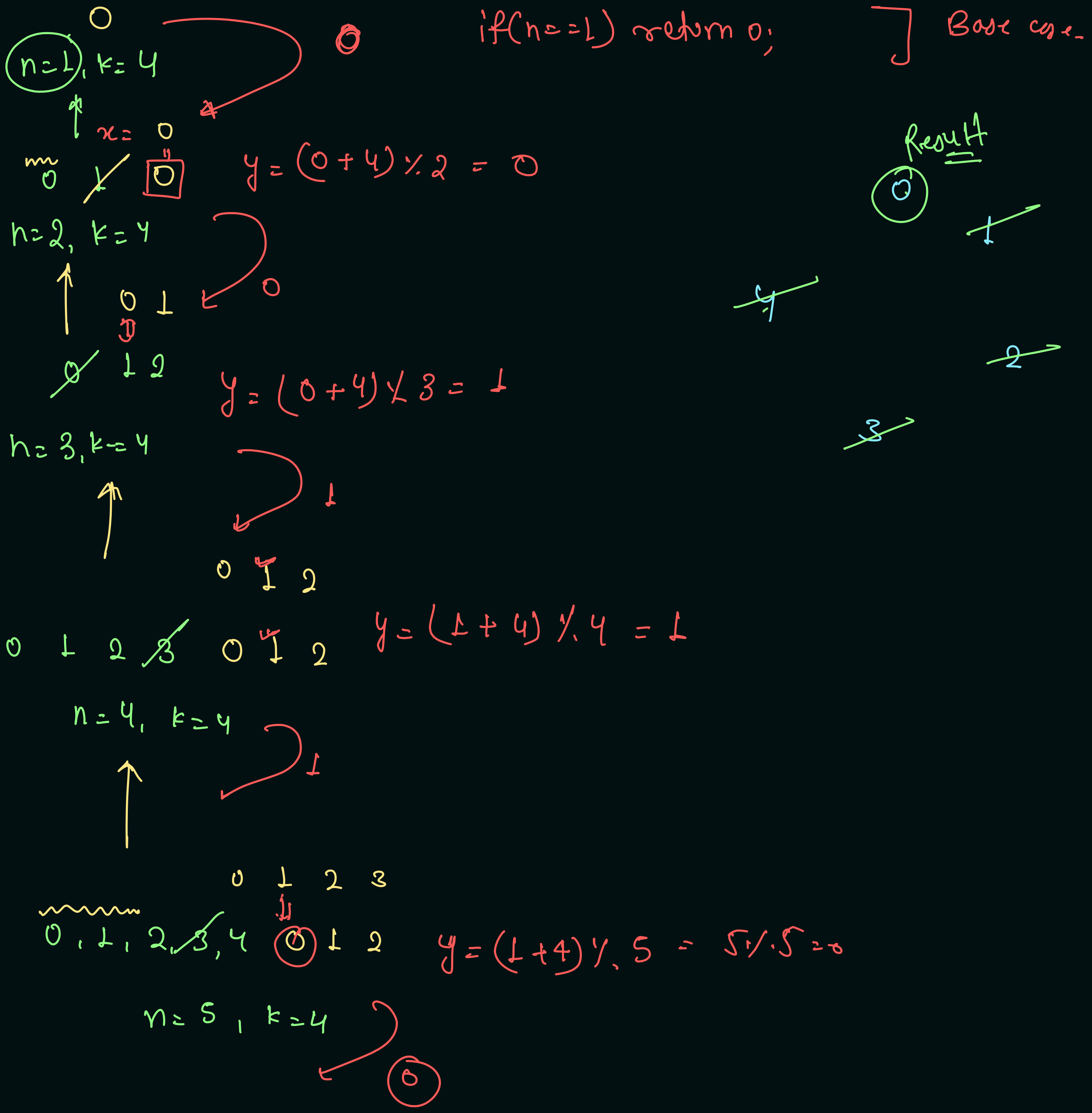
faith → $x = (n-1, k)$ → send you no. of remaining sure.

Merging → find 'y' & return to next level

return = $(x+k) \% n$;

n=5, k=4.

```
p. int fun(int n, int k) {

  if(n == 1)
      return 0;

  int x = fun(n-1, k);

  retur (x+k) % n;

}
```



if(n==1) return 0;] Base case.

$y = (0+4) \% 2 = 0$

$y = (0+4) \% 3 = 1$

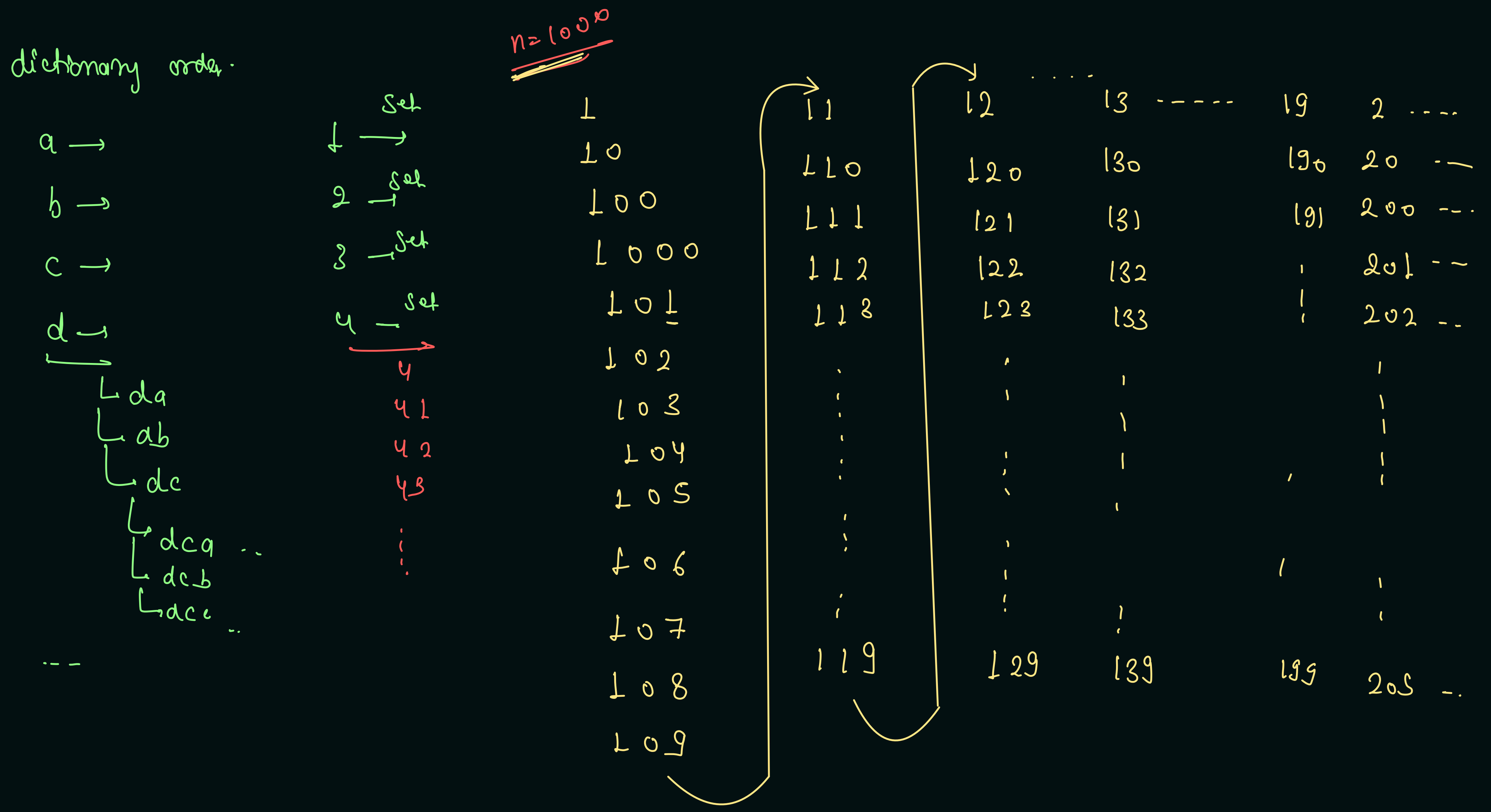
$y = (1+4) \% 4 = 1$

$y = (1+4) \% 5 = 5 \% 5 = 0$

lexicographical printing: n= 1000   1 to 1000] print no. in lexicographical order

order of dictionary

dictionary order.



a →

b →

c →

d →

da

ab

dc

dca ..

dcb

dcc ..

---

1 → set

2 → set

3 → set

4 → set

4

41

42

43

⋮

n=1000

1
10
100
1000
101
102
103
104
105
106
107
108
109

11
110
111
112
113
⋮
119

12
120
121
122
123
⋮
129

13
130
131
132
133
⋮
139

19
190
191
⋮
199

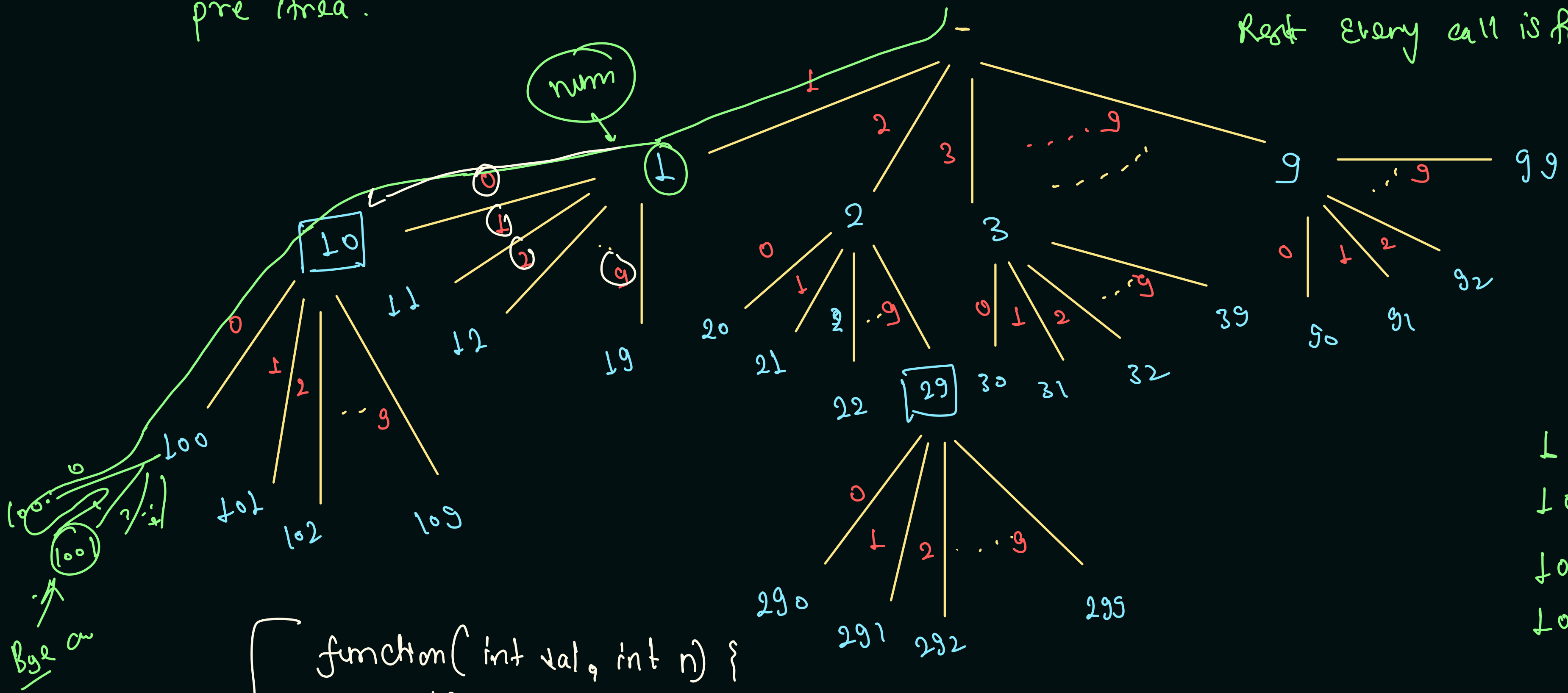
2
20
200
201
202
⋮
205

Note → answer will print in pre Area.

n = 1000

first level call from main function → from 1 to 9.

Rest Every call is from 0 to 9



1
10
100
1000

Solution

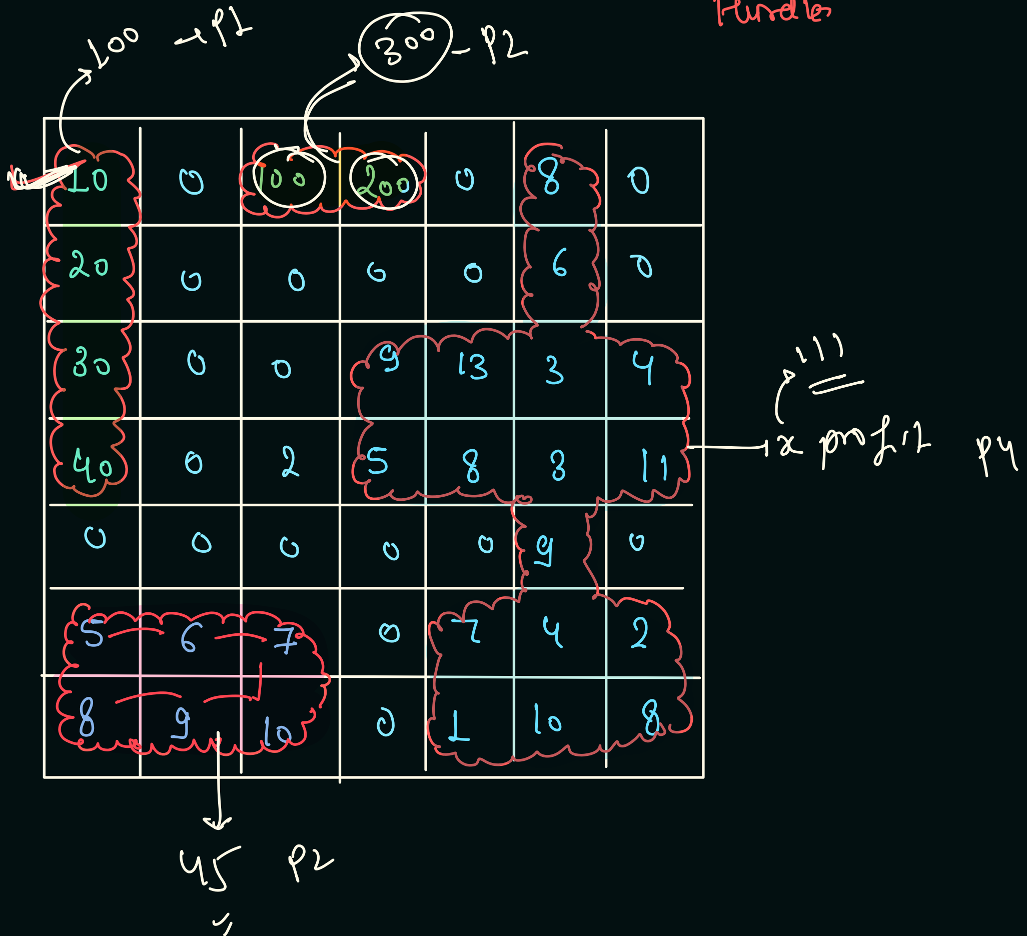
```
function( int val, int n) {
        if ( val > n)  return;

        syso( val);
        for(int i=0; i<=9; i++){
                function( 10 * val + i, n);
        }
}
```

Goldmine2:

if (mine[i][j]==0) → Barrier/ Hurdles

connected from top-left-down-right

mine→



10	0	100	200	0	8	0
20	0	0	0	0	6	0
30	0	0	9	13	3	4
40	0	2	5	8	3	11
0	0	0	0	0	9	0
5	6	7	0	7	4	2
8	9	10	0	1	10	8

max profit =?

Result= max(P1, P2, P3, P4)

Solve Sudoku:

number will no Repeat in
① same Row
② same col
③ same submatrix

	0	1	2	3	4	5	6	7	8
0	0	3	0	1	5	6	0	0	0
1	0	8	0	0	2	0	0	7	0
2	6	0	0	0	0	0	5	0	0
3	0	1	0	6	0	0	9	0	0
4	2	0	0	9	4	1	0	0	6
5	0	0	8	0	0	5	0	1	0
6	0	0	7	0	0	0	0	0	9
7	0	5	0	0	1	0	0	8	0
8	0	0	0	2	6	8	0	4	0

Safety: → ~~①~~ same Row
~~②~~ same col
~~③~~ same submatrix

Starting of Submatrix →

rr = r - r%3

cc = c - c%3

Eg ⇒ 5, 8

5 - 5%3 = 5-2 = 3

8 - 8%3 = 8-2 = 6

```
isSafe to place( int i, int j, int num, int[][] board){
    // same col ⟶ loop in Row
    // same Row ⟶ loop in col,
    // for sub matrix
    int rr = i - i%3 ;
    int cc = j - j%3
}
```

```
for(int r=0; r<3; r++){
    for(int c=0; c<3; c++){
        int ii = r+rr;
        int jj = c+cc;
        if(bd[ii][jj] == num) return false
}
```

After all check → return 0;

	0	1	2	3	4	5	6	7	8
0 →	0 0	3 1	0 2	1 3	5 4	6 5	0 6	0 7	0 8
1	0 9	8 10	0 11	0 12	2 13	0 14	0 15	7 16	0 17
2	6 18	0 19	0 20	0 21	0 22	0	5	0	0
3	0	1	0	6	0	0	9	0	0
4	2	0	0	9	4	1	0	0	6
5	0	0	8	0	0	5	0	1	0
6	0	0	7	0	0	0	0	0	9
7	0	5	0	0	1	0	0	8	0
8	0	0	0	2	6	8	0	4	0

2D in 1D.

Array list of cell number →

0	2	6	7	8	9	11	12	14	15	. . . . .

Recursion on list → $\underline{0}$ → cell no.

find r, c

r = cell / Total col

c = cell % col

How to find cell no :

cell no = n * r + c

r=0
c=5

9*0+5 = $\boxed{5}$

r=1
c=3

9x1+3 = $\boxed{12}$

if cell no = 7

$\cancel{r}$ = 7/9 = 0

$\cancel{c}$ = 7 % 9 = 7

```
for(int num = 1 to 9; n++)
    check safety,
    bd[r][c] = num;
    call
    bd[r][c] = 0;
}
```