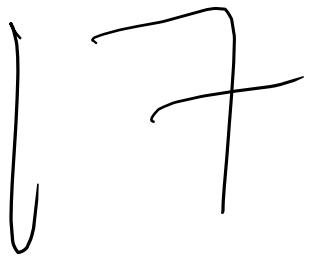
Let’s solve it



Count positive (+ve) (>0) inputs; stop at zero.

Ignore negative.

```
int n; int count=0;
do
{scanf("%d", &n);
 if( n>0)
  { count++;
```

```
while(n!=0);
printf("%d", count);
```

```
while(1)
{ scanf("%d", &n);
  if(n==0)
    break;
  if(n<0)
  { continue;
    count++;
  }
printf("%d", count);
```

```
for( _ ; _ ; )
{
    if( ——— )
    {
        break;
    }
}
```

```
while(—)
{
    if( ——— )
    {
        continue;
    }
}
```

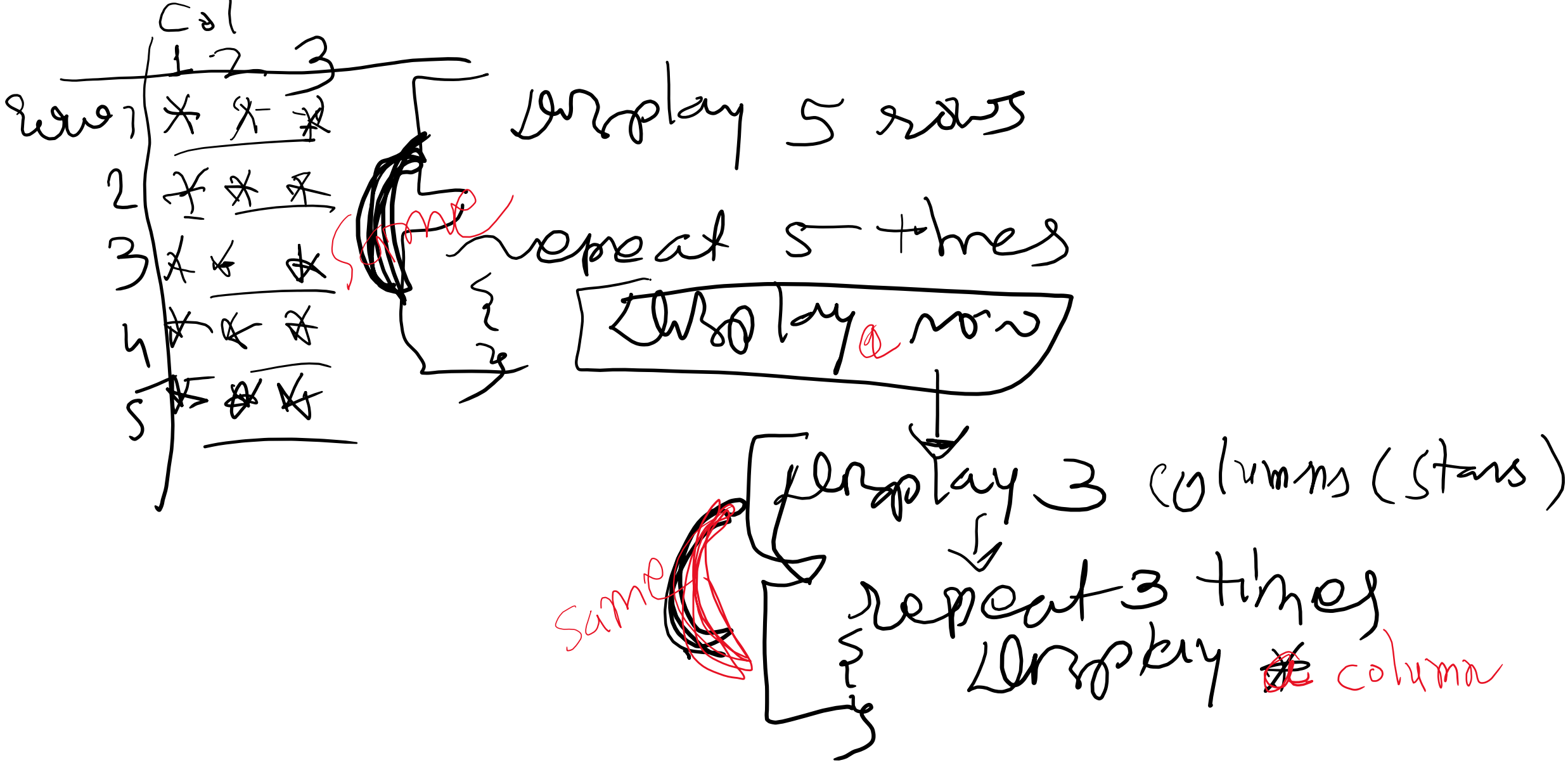
Ex 1

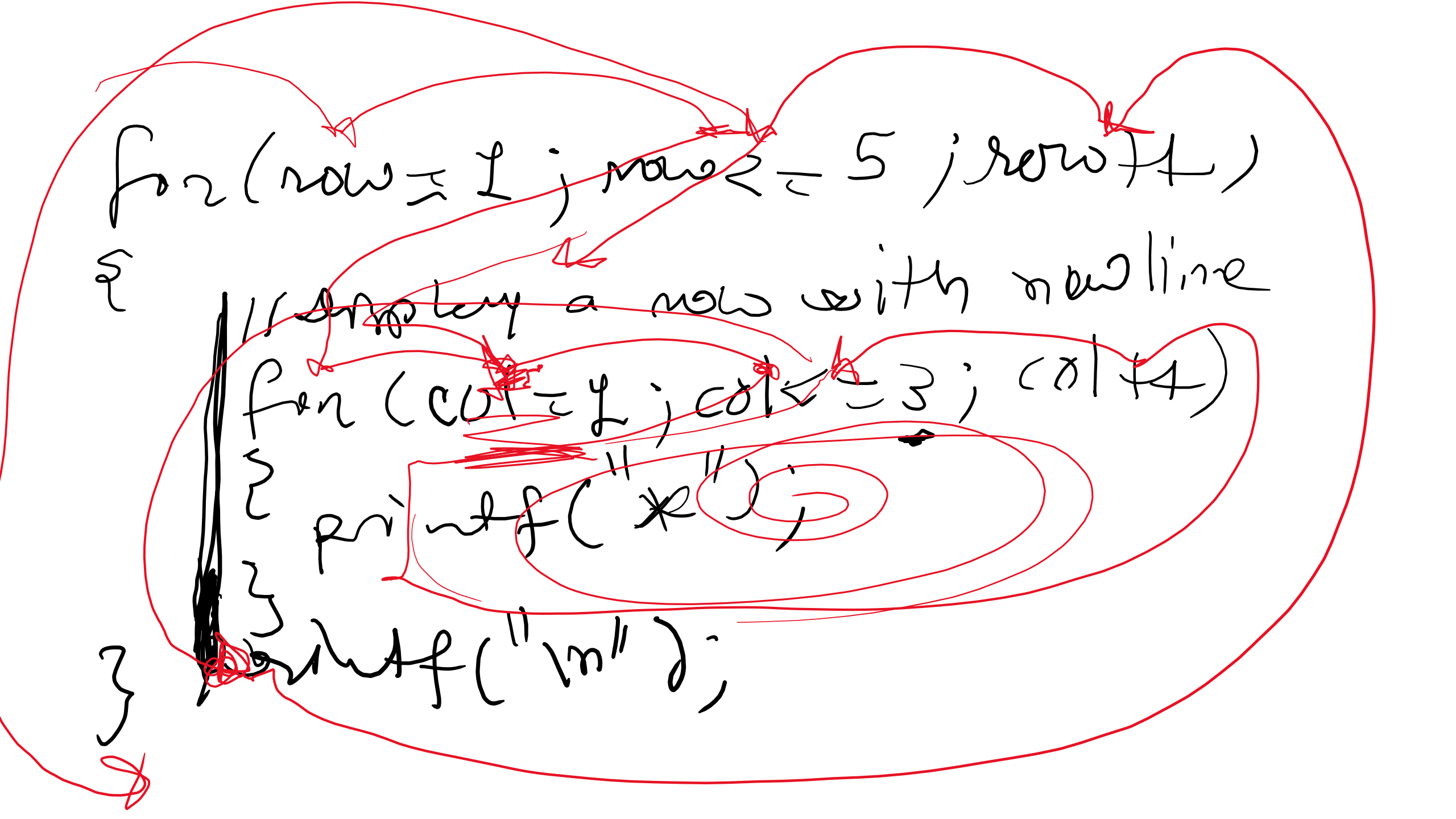
nesting of for loops

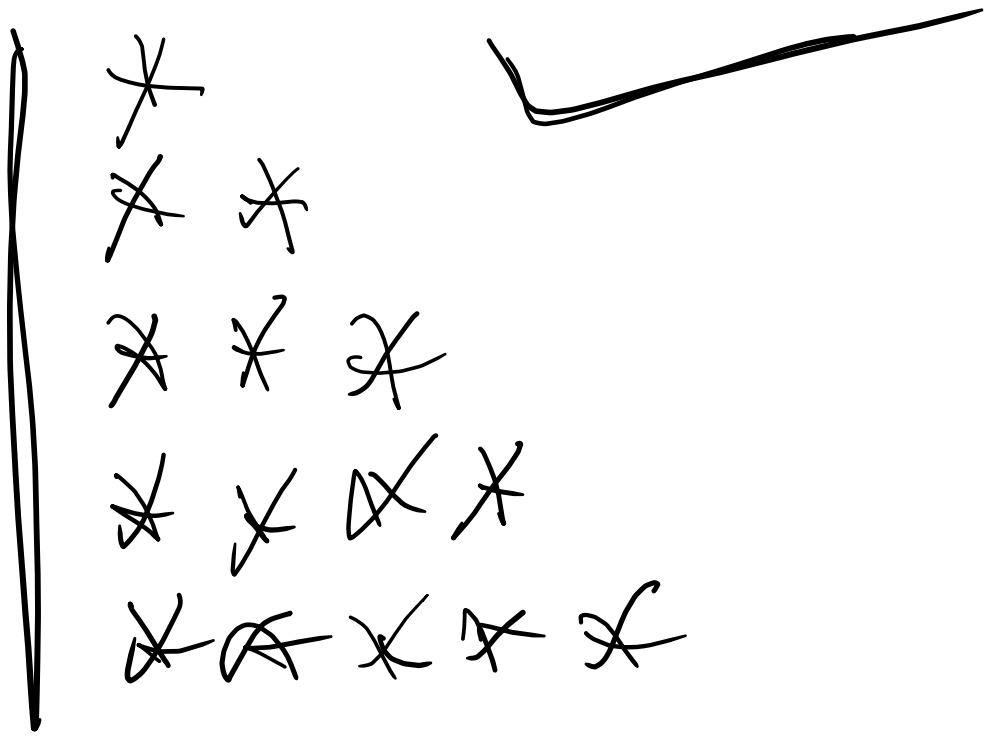
```
111
222
333
444
```

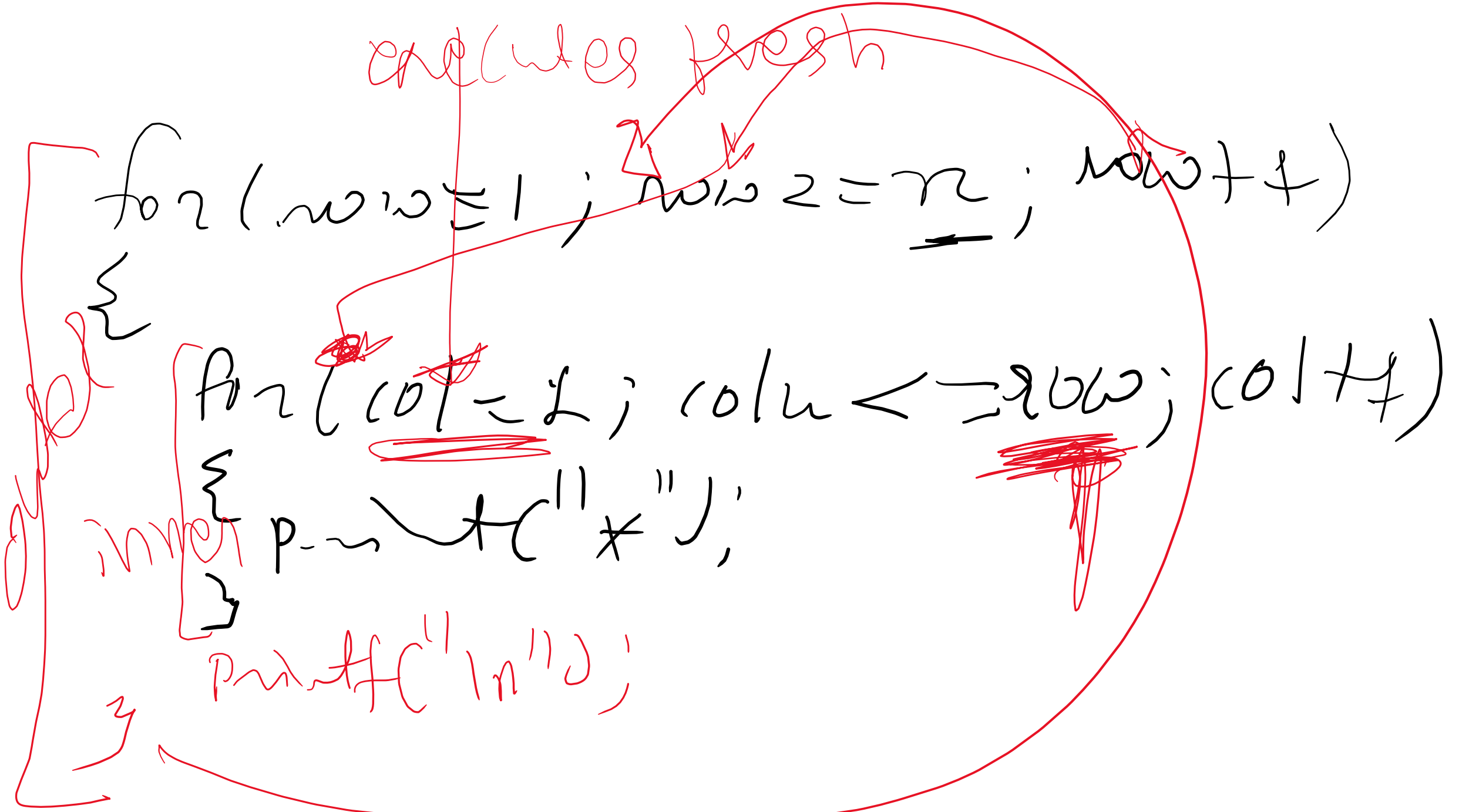
```
123
456
789
```

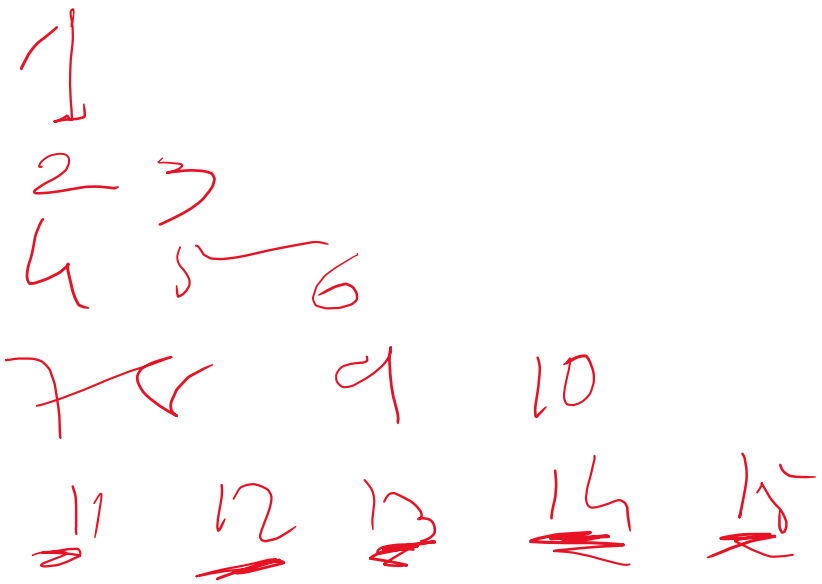
10 11 12
13 14 15


Col
1 2 3
Row
1
2
3
4
5
same
display 5 rows
repeat 5 times
{
}
display a row
display 3 columns (stars)
same
repeat 3 times
{
}
display * column


for(row=1;row<=5;row++)
{
// display a row with newline
for(col=1;col<=3;col++)
{
printf("*");
}
printf("\n");
}




for(row=1; row<=n; row++)
{
for(col=1; col<=row; col++)
{
p-~+("*");
}
printf("\n");
}




```
int data[5][5];
for(row=1; row<=5; row++)
{
    for(col=1; col<=5; col++)
    {
        printf("%d", data[row][col]);
    }
    printf("\n");
}
```

1

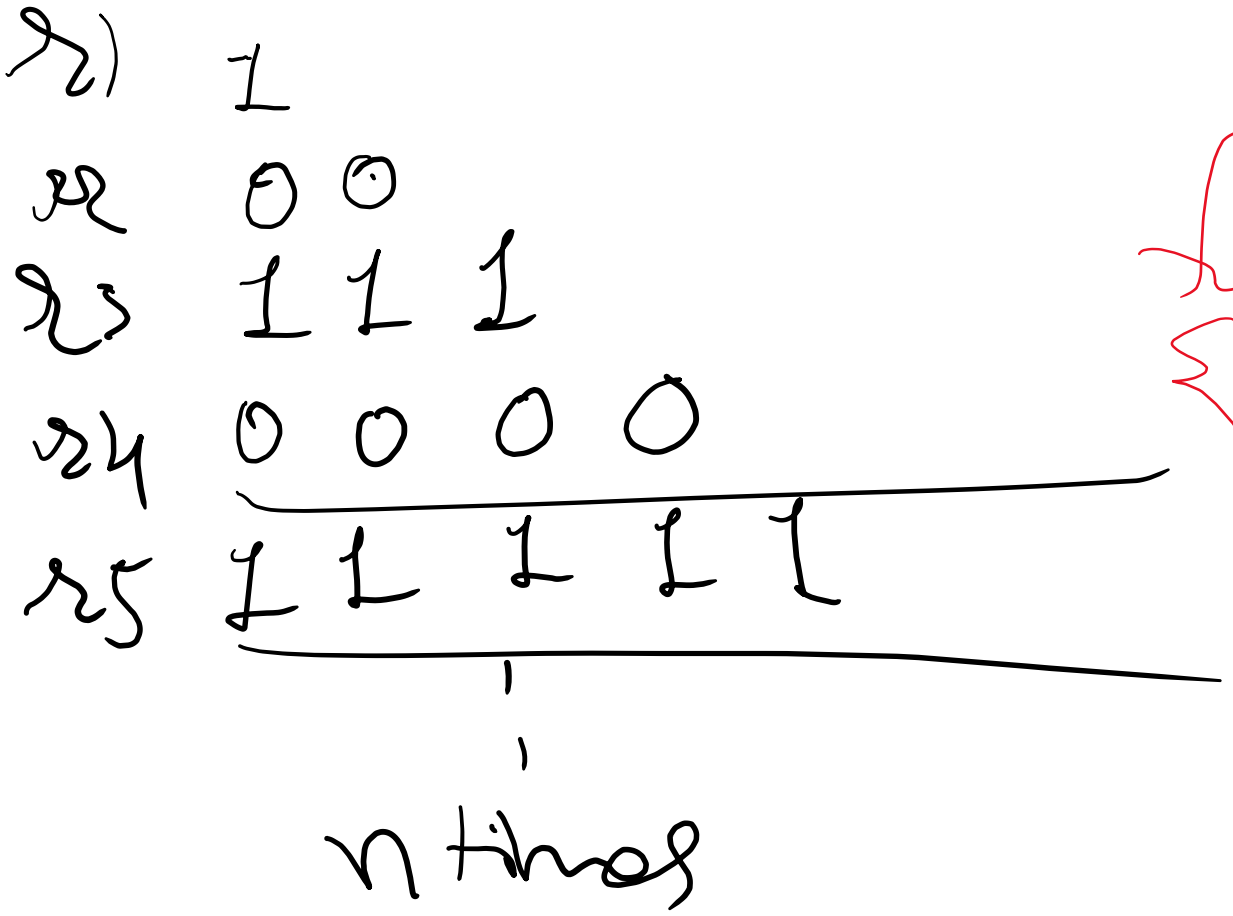
4 4

9 9 9

16 16 16 16

25 25 25 25 25

printf("%d", sqno*row);




```
for(row=1
{
for(col=1
{
if((row+col)%2==0)
printf("1");
else
printf("0");
}
}
```

Σ

A
B1 } → B
B2
C

// decreases

// increases

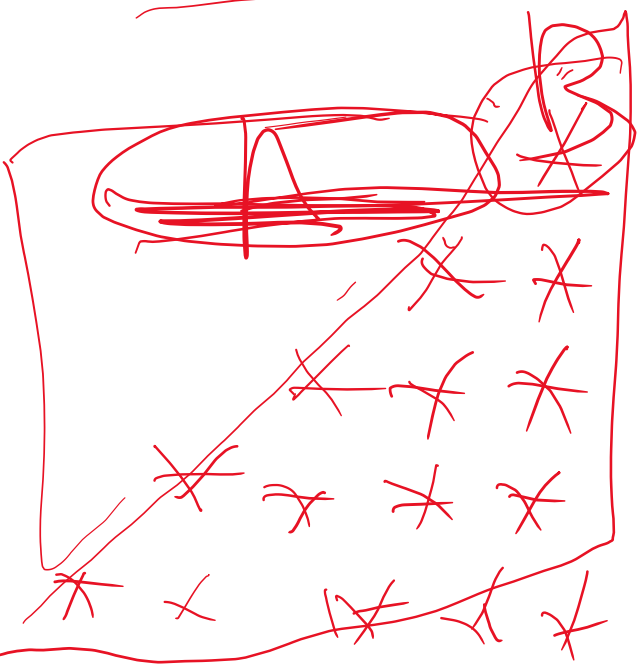
// overall increases

∫

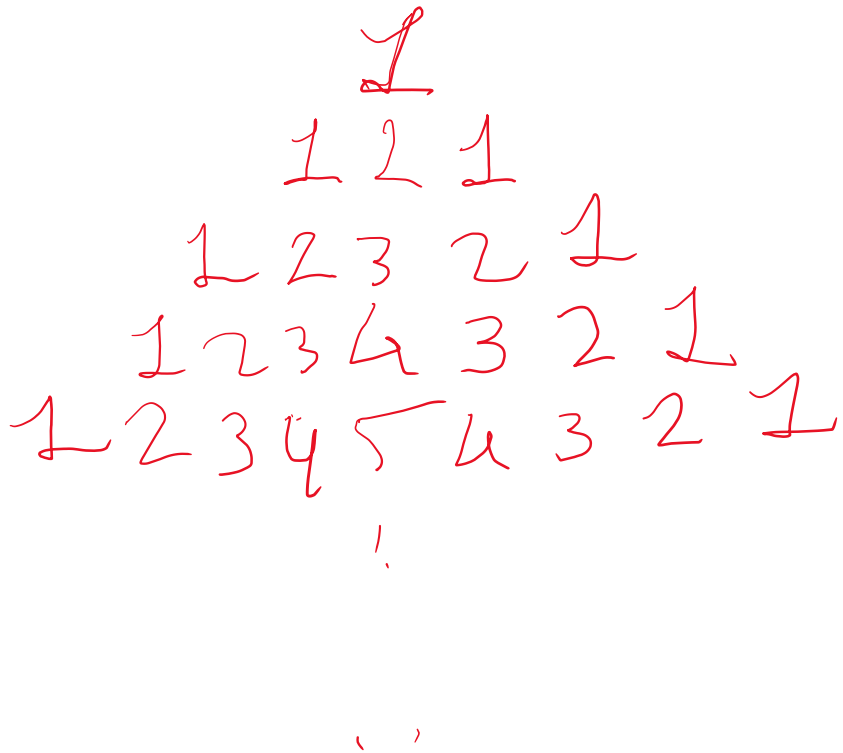
```
  1
 0 1
-----
 0 1 0
-------
 1 0 1 0
1 0 1 0 1
```

```
   1
  0 1
-------
  1 0 1
-------
 0 1 0 1
1 0 1 0 1
```

Hint: use a flag (0/1) variable.



for( )
{
LA for space
LB for symbols
}


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