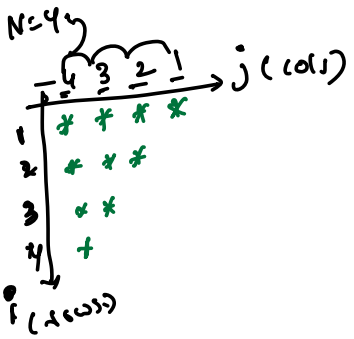
Q Give N. Print the following pattern (Inverted triangle)

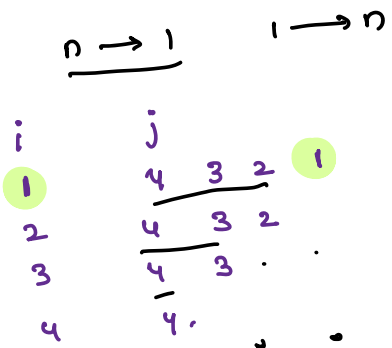
N = 4

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

N = 3

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





i <= j => j >= i

4 → 3 → 2 → 1.

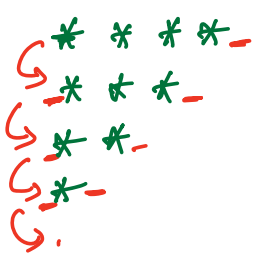
j++ or j--

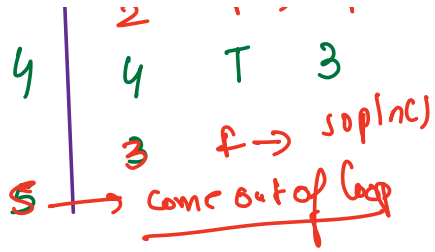
```
for (int i=1; i<=N; i++) {
    for (int j=N; j>=i; j--) {
        System.out.print("*");
    }
    System.out.println();
}
```

O(n²)

N = 4

i	j	j >= i	j--
1	4	T	3
	3	T	2
	2	T	1
	1	T	0
	0	F	println()
2	4	T	3
	3	T	2
	2	T	1
	1	F →	println()
3	4	F	3
	3	T	2
		F →	println()





Q Patterns with spaces

— → spaces

N = 3

outer

```
* - - *
* - - *
* - - *
```

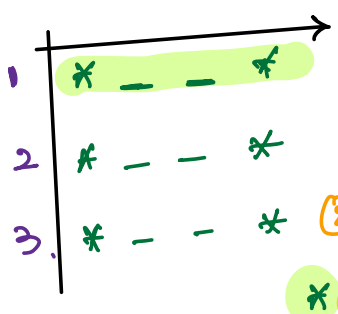
N = 4

```
* - - - *
* - - - *
* - - - *
* - - - *.
```

5 → 4 spaces

6 → 5 spaces

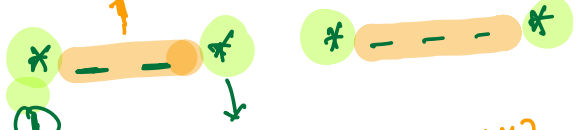
(N-1) spaces



loop 1 for

for (1 — N)

② print (N-1) spaces



① sopc(*) ← print 1 star

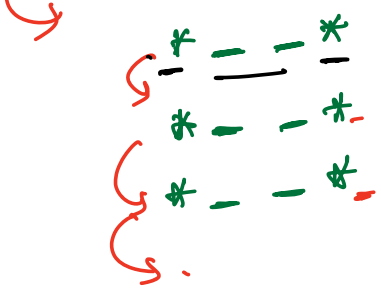
③ print 1 star → sop(*)

1 →

```
* - - *
* - - *
* - - *
```

```
for (int i=1; i<=N; i++) {
    sop ('*');
    for (int j=1 ; j<= N-1 ; j++) {
        sop (" _ ");
    }
    sop ("*");
    sopln ();
}
```

```
for(int i=1; i<=N; i++){
    sop('*');
    for(int j=1; j<=N-1; j++){
        sop("_");
    }
    sop("*");
    soPln();
}
```



i	'*'	j	j<=N-1	j++	* sop
1	P	1	T	2	
		2	T	3	
		3	F →		P sopln()
2	P	1	T	2	
		2	T	3	
		3	F →		P sopln()
3	P	1	T	2	
		2	T	3	
		3	F →		P sopln()
4 → come out of loop					

B) Print the following pattern.

N=3

```
*--*
*-*
**
```

N=4

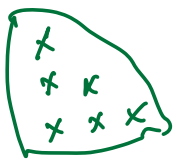
```
*---*
*--*
*-*
**
```

N=5

```
*----*
*---*
*--*
*-*
**
```

```
for(int i=1; i<=N; i++){
    sop('*');
    for(int j=1; j<=N-1; j++){
        sop("_");
    }
    sop("*");
    soPln();
}
```

Q Pl → Print the following pattern.



j <= i

N = 3

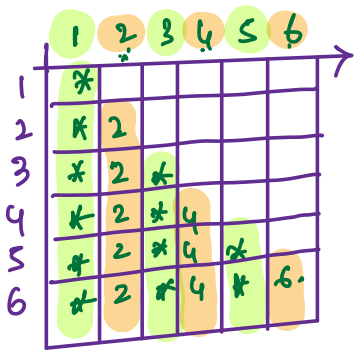
```
*
* 2
* 2 *
```

N = 5

```
*
* 2
* 2 *
* 2 * 4
* 2 * 4 *
```

N = 6

```
*
* 2
* 2 *
* 2 * 4
* 2 * 4 *
* 2 * 4 * 6.
```



→ 1, 3, 5 → *

2, 4, 6 → number → j

j j j

```
for(int i=1 ; i<=n; i++){
    for(int j=1; j<=i; j++){
        if(j%2==1){
            sop('*');
        }else{
            sop(j);
        }
    }
    sop(nc); Move to next line
}
```

come out

N = 4

```
*
* 2
* 2 *
* 2 * 4
```

i	j
1	1
	2 → breakln
2	1
	2
	3 → breakln
3	1
	2
	3
	4 → breakln
4	1
	2
	3
	4
	5 → breakln

5 → break - come out

Break 10:16 → 10:27

Sponsored by (Engineers) Clinic

Q

N = 3

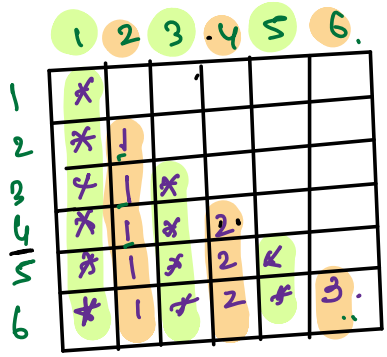
```
*
* 1
* 1 *
```

N = 5

```
*
* 1
* 1 *
* 1 * 2
* 1 * 2 *
```

N = 6

```
*
* 1
* 1 *
* 1 * 2
* 1 * 2 *
* 1 * 2 * 3.
```



1, 3, 5 → * (odd value of j)

2, 4, 6 → num → j/2 (even value of j)

j = 2/2 → 1 .. j/2
j = 4/2 → 2 j/2
j = 6/2 → 3 j/2

```
for(int i=1 ; i<=n; i++){
    for(int j=1, j<=i; j++){
        if( j % 2 == 0 ){ → even numbers
            sop(j/2);
        }else{
            sop("*");
        }
    }
    sop(nc);
}
```

2nd approach

Variable counter → count = 0;

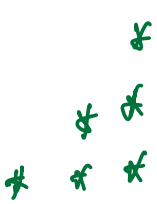
```
*
* 1
* 1 *
```

```
for(int i=1 ; i<=n; i++){
    int count=1;
    for(int j=1; j<=i; j++){
        if(j%2==0){ → even numbers
            sop(count);
            count++;
        }else{
            sop("*");
        }
    }
    sop(nc);
}
```

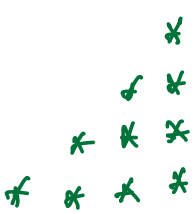
N=9

i	j	count
1	1	1
	2 → break in	
2	1	1
	2 →	2
	3 → break in	
3	1	1
	2 →	2
	3	2
	4 → break in	
4	1	1
	2 →	2
	3	2
	4 →	3
5 → Come out of loop		

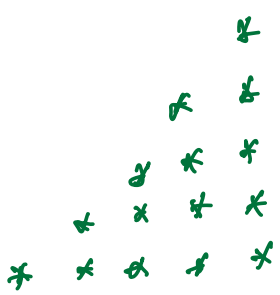
Q N=3



N=4

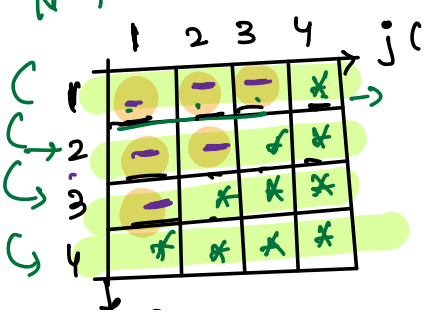


N=5



outloop

N=4



1st row
3 spaces + 1 star

i	spaces	stars
1	3 (4-1)	1
2	2 (4-2)	2
3	1 (4-3)	3
4	0 (4-4)	4

i (rows)

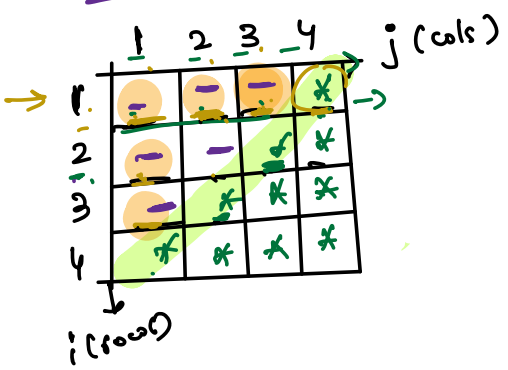
1 — (n-i) spaces + (i) stars (n-i) (i)

for + for.

(spaces) + (stars) →

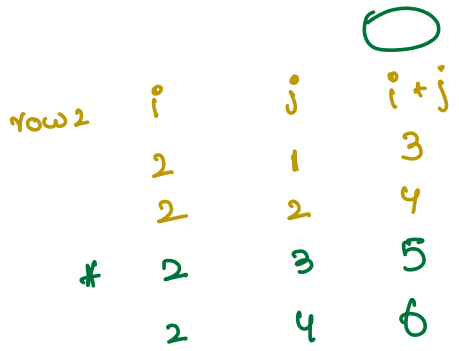
```
for(int i=1 ; i<=n; i++){
    // spaces
    for(int j=1; j<=n-i ; j++){
        sop("_");
    }
    // stars
    for(int j=1 ; j<=i ; j++){ → compilation error
        sop("*");
    }
    sopln();
}
```

2nd approach



N = 4

i	j	(i + j)	N+1
1	4	5	5
2	3	5	5
3	2	5	5
4	1	5	5.




```
for(int i=1; i<=n; i++){
    for(int j=1; j<=n; j++){
        if(i+j<n+1){
            sop("_");
        }
        else{
            sop("*");
        }
    }
    soplnc);
}
```

https://www.interviewbit.com/snippet/8f62783963837610fd3e/

Doubt session

6 →

, 6

perfect number — whose sum of divisors is 6

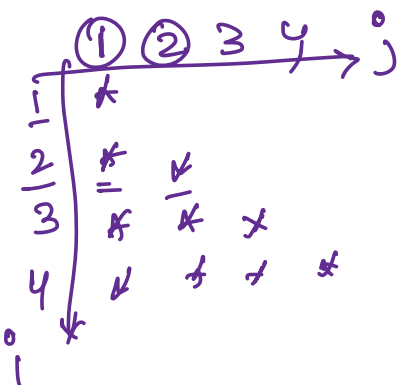
Prime number
1 divisible by 1 and only itself.

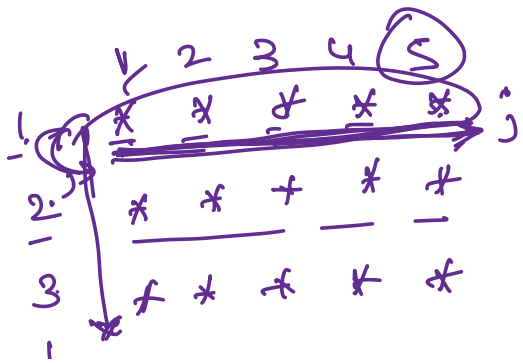
6 → 1, 2, 3 →

sum = 1 + 2 + 3 → 6

6 == 6 → perfect number.

```
while(T>0){
  int num = sc.nextInt();
  int sum = 0;
  for(int i=1; i<=N; i++){
    if(N%i==0){
      sum = sum + i;
    }
  }
  if(sum==num){
    sop("num");//Perfect.
  }
}
```





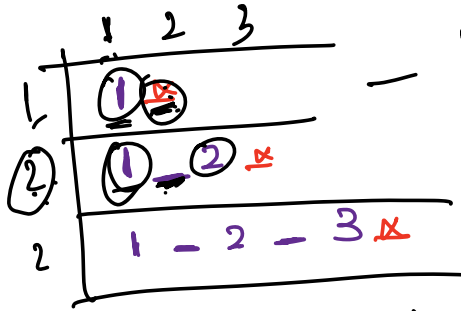
```
for(int i=1 ; i<=4 ; i++)
    for(int j=i ; j<=i ; j++)
        sop(j);
    sop(inc);
```

```
for(int i=1; i<=3 ; i++){
    for(int j=1; j<=5 ; j++){
        sop(*);
    }
    sop(inc);
}
```

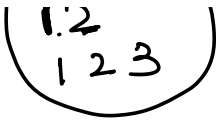
i j

1 — 1
2 — 1 2
3 — 1 2 3
4 — 1 2 3 4

$j <= i \quad || \quad i >= j$

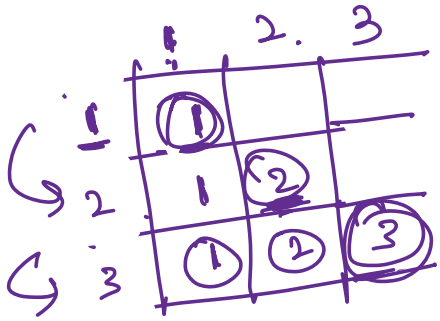



```
for(int i=1 ; i<=3 ; i++)
    for(int j=1 ; j<=i ; j++)
        sop(j+" ");
```



1
1 - 2
1 - 2 - 3 .

1 2 3




```
sop(j)2
if(i!=j){
  sop("-");
}
```

```
(i==j)
if(i!=j)
  print("-");
{
```

i	j	i!=j
1	1	f
1	2	T
2	1	T
2	2	f
3	1	T
3	2	T
3	3	f

1
1 - 2 *
1 - 2 - 3 *

N = 5 => 5! = 5 x 4 x 3 x 2 x 1

= 120

4! = 4 x 3 x 2 x 1

= 24.

```
int fact=1;
for(int i=N; i>=1; i--){
    fact = fact * i;
}
sop(fact)
```

1 x 2 x 3 x 4