

① input: 6  
output: 1 2 3 4 5 6

```
{  
    input;  
    for(..<input..)  
        print();
```

② input: 9  
6  
↓..  
output: 123 456  
123456  
123456  
123456

```
row=9    col=6  
for(int i=1; i<=row; i++){  
    for(int j=0; j<col; j++){  
        sop(j+" ");  
    } sopln;
```

③ input: 4  
output: 1  
12  
123  
1234

```
line=4; 2  
for(int i=1; i<=line; ++i){  
    for(int j=1; j<=i; ++j)  
        sop(j+" ");  
    sopln();
```

④ input: 9  
output: 9  
3 9  
2 3 9  
1 2 3 9  
line=9;  
for(int i=1; i<=line; ++i){  
 for(int j=1; j<=line-i; ++j) {sop(" ");}  
 for(int k=line-i+1; k<=line; ++k) {sop(k);}  
 sopln();

⑤ input: 4  
output: 1  
12  
123  
1234  
lines=4  
for (int i=1; i<=lines; ++i){  
 for (int j=1; j<=lines-i; ++j) {sop(" ");}
 for (int k=1; k<=i; ++k) {sop(k);}
 sopln();

⑥ input: 3  
output: 1  
123  
12345  
123456  
lines=3  
i=1 3-1 3-1  
i=2 -1 2 3 -  
i=3 1 2 3 4 5  
5 (2n-1)  
odd numbers  
for (int i=1; i<=lines; ++i){  
 for (int j=2; j<=lines-i; ++j) {sop(" ");}
 for (int k=1; k<=2i-1; ++k) {sop(k);}
 sopln();

⑦ input: 3  
output: 1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5  
lines=3  
upper half  
bottom half  
for (int i=lines-1; i>=1; --i){  
 sopln();

⑧ input: 9  
5  
output: 12345  
1 5  
1 5  
1 2 3 4 5

⑨ input: 5

output:  
1  
1 2  
1 3  
1 2 3 4  
1 2 3 4 5

⑩ input: 5

output:  
5  
4 5  
3 5  
2 5  
1 2 3 4 5

⑪ input: 3

output:  
1  
1 3  
1 2 3 4 5

⑫ input: 3

output:  
1  
1 3  
1 2 5  
1 3  
1

⑬ input: 5

output: 123454321

⑭ input: 5

output:  
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

⑮ input: 6

output: \* \* \* \* \* \*

(16) input: 4  
6

output:

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

(17) input: 4

output:

```
*
```

```
* *
```

```
* * *
```

```
* * *
```

(18) input: 4

output:

```
*
```

```
* *
```

```
* * *
```

```
* * *
```

(19) input: 3

output:

```
*
```

```
* *
```

```
* * *
```

(20) input: 3

output:

```
*
```

```
* *
```

```
* * *
```

```
* * *
```

```
* *
```

```
*
```

(21) input: 4  
5

output:

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

(22) input: 5

output:

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

(23) input: 5

output:

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

(24) input: 3

output:

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

(25) input: 3

output:

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