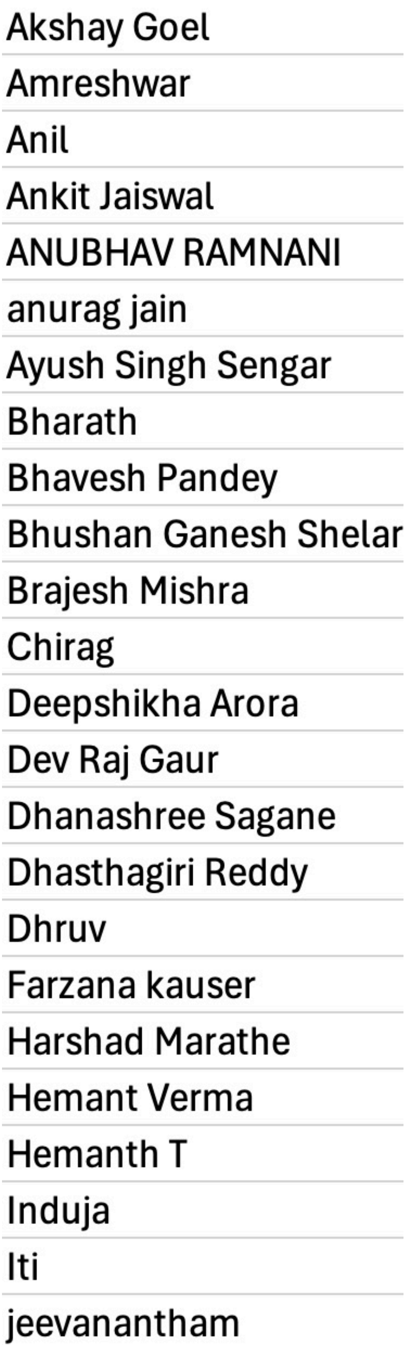
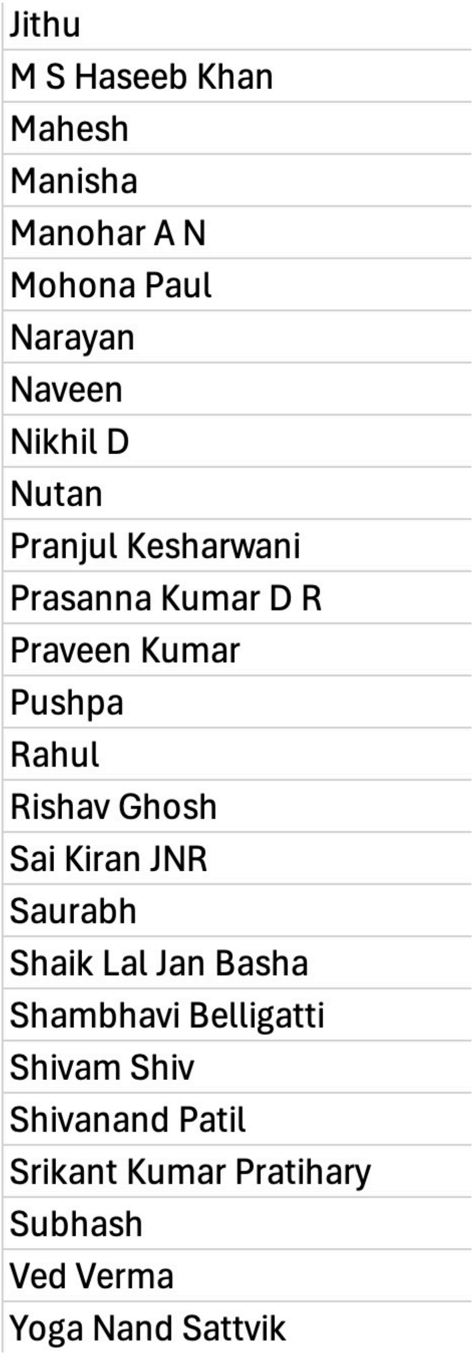
Patterns



Akshay Goel
Amreshwar
Anil
Ankit Jaiswal
ANUBHAV RAMNANI
anurag jain
Ayush Singh Sengar
Bharath
Bhavesh Pandey
Bhushan Ganesh Shelar
Brajesh Mishra
Chirag
Deepshikha Arora
Dev Raj Gaur
Dhanashree Sagane
Dhasthagiri Reddy
Dhruv
Farzana kauser
Harshad Marathe
Hemant Verma
Hemanth T
Induja
Iti
jeevanantham



Jithu
M S Haseeb Khan
Mahesh
Manisha
Manohar A N
Mohona Paul
Narayan
Naveen
Nikhil D
Nutan
Pranjul Kesharwani
Prasanna Kumar D R
Praveen Kumar
Pushpa
Rahul
Rishav Ghosh
Sai Kiran JNR
Saurabh
Shaik Lal Jan Basha
Shambhavi Belligatti
Shivam Shiv
Shivanand Patil
Srikant Kumar Pratihary
Subhash
Ved Verma
Yoga Nand Sattvik

Q1> Given N as input, print * N times

I : 5

O : * * * * *

```
public static void main (.... ) {
        // Initialise scanner .....
        int N = scanner.nextInt();

        for (int t = 1 ; t <= N ; t++) {

                System.out.print("*");
        }
}
```

Q2> Given N as input. Print a square of N*N size containing * in each cell

I : 5

O :

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

```
public static void main (.... ) {
        // Initialise scanner .....
        int N = scanner.nextInt();
        for (int r = 1 ; r <= N ; r++) {
```

```
        for (int c = 1 ; C<=N ; c++) {
            System.out.print("*") ;
        }
        System.out.println() ;
    }
}
```

Q3> Given input N, M . Print a rect. of N*M size containing * in each cell

I : N= 3 M = 4

O :

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

```
public static void main (....) {
    // Initialise Scanner .....
    int N = scanner.nextInt() ;
    int M = scanner.nextInt() ;
    for (int r = 1 ; r <= N ; r++) {
        for (int c = 1 ; C<=M ; c++) {
            System.out.print("*") ;
        }
        System.out.println() ;
    }
}
```

r	c
1	[1,2,3,4]
2	[1,2,3,4]
3	[1,2,3,4]

Q4> Given N as input, print staircase of size N

I: 5

O:

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

Row	Stars
1	1
2	2
3	3
4	4
5	5

```
public static void main (....) {
    // Initialise Scanner .....
    int N = scanner.nextInt();
    for (int r = 1 ; r <= N ; r++) {
        for (int c = 1; c<=r; c++) {
            System.out.print("*");
        }
        System.out.println();
    }
}
```

r	c
1	[1]
2	[1 2]
3	[1 2 3]
4	[1 2 3 4]
5	[1 2 3 4 5]

Q5> Given N as input . Print the pattern as shown

I: 3

O:

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

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

Dry Run

I: 4

O:

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

```
1   *
2   *  2
3   *  2  *
4   *  2  *  4
    1  2  3  4
```

Dry run

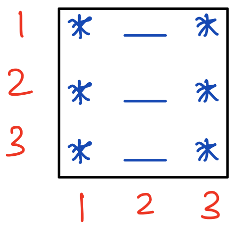
```
public static void main (....) {
        // Initialise Scanner .....
        int N = scanner.nextInt();
        for (int r = 1 ; r <= N ; r++) {
            for (int c = 1 ; C <= r ; c++) {
                if ( c%2 == 0 ) { // even
                        system.out.print(c);
                } else {
                        System.out.print("*");
                }
            }
            System.out.println();
        }
}
```

N = 3

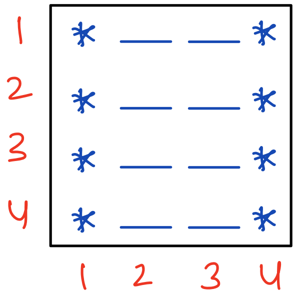
r	c	
1	[1]	*
2	[1 2]	* 2
3	[1 2 3]	* 2 *

Q6> Given N as input, print the pattern as shown

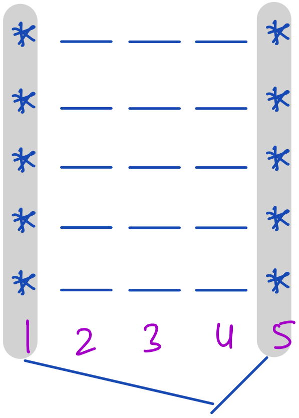
N= 3



N= 4



N=5



For the first and last column print *
otherwise print _

```
public static void main (.... ) {
        // Initialise Scanner .....
        int N = scanner.nextInt() ;
        for (int r = 1 ; r<=N ; r++) {
                for (int c = 1 ; C<=N ; c++) {
                        if ( c==1 || c==N) {
                                System.out.print("*") ;
                        }
                        else { System.out.print("_") ;}
                }
                System.out.println() ;
        }
}
```

Q7> Given N as input, Print the pattern as shown

N = 3

	r	stars
* * *	1	3
* *	2	2
*	3	1

N = 4

	r	stars
* * * *	1	4
* * *	2	3
* *	3	2
*	4	1

```
public static void main (....) {
        // Initialise Scanner .....
        int N = scanner.nextInt(); // N=3
        for (int r = 1 ; r <= N ; r++) {
            for (int c = N ; c >= r ; c--) {
                System.out.print("*") ;
            }
            System.out.println() ;
        }
}
```

r	c		
1	3	2	1
2	3	2	
3	3		

```
f r ⟶ [1 to N] {
    f c ⟶ [r to N] {
        print(*)
```

```
f r ⟶ [1 to N] {
    count = N-r+1
    for c ⟶ [1 to count]
        print (*)
```

Break 22:35

Q8> Given N as input, Print the pattern as shown

N = 3 N = 4

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

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

r	spaces	stars
1	2	2
2	1	2
3	0	2

N=3

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

r	spaces	stars
1	$\overset{3-1}{N-r} = 2$	2
2	$\overset{3-2}{N-r} = 1$	2
3	$\overset{3-3}{N-r} = 0$	2

For every row do the below

- * loop N-r time and print space *

```
public static void main (....) {
        // Initialise scanner .....
        int N = scanner.nextInt(); // N=3
        for (int r = 1 ; r <= N ; r++) {
                System.out.print("*");
                for (int c = 1 ; c <= N-r ; c++){
                        system.out.print("  ");
                }
                System.out.print("*");
                system.out.println();
        }
}
```

N=4

r	N-r	c	print
1	3	[1 2 3]	* s s s *
2	2	[1 2]	* s s *
3	1	[1]	* s *
4	0	[]	**

Q9> Given N as input, Print the pattern as shown

N = 3

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

N = 4

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

N = 4

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

r	spaces	stars
1	3	1
2	2	2
3	1	3
4	0	4

For every row

→ print <u>N-r</u> spaces first

→ print <u>r</u> stars

```
public static void main (....) {
        // Initialise Scanner .....
        int N = scanner.nextInt(); // N=3
        for (int r = 1 ; r<=N ; r++) {
                // Print N-r spaces
                for (int c = 1 ; c<= N-r ; c++){
                        System.out.print("  ");
                }
                // Print r stars
                for (int c = 1 ; c<= r ; c++){
                        System.out.print("*");
                }
                System.out.println();
        }
}
```

N = 4

r	N - r	c
1	3	1 2 3
2	2	1 2
3	1	1
4	0	

second loop

r	c	
1	1	S S S *
2	1 2	S S * *
3	1 2 3	S * * *
4	1 2 3 4	* * * *

Q10> Given N as input, Print the pattern as shown

N = 4

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

N = 4

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

r	left*	spaces	right*
1	4	0	4
2	3	2	3
3	2	4	2
4	1	6	1

N = 4

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

r	left*	spaces	right*
1	N-r+1		N-r+1
2		(r-1)*2	
3			
4			

```
for every row
    print N-r+1 stars
    print 2*(r-1) spaces
    print N-r+1 stars
```

```
public static void main (....) {
    // Initialise Scanner .....
    int N = scanner.nextInt(); // N=3
    for (int r = 1 ; r <= N ; r++) {
        // Print N-r+1 stars
        for ( int c = 1 ; c <= N-r+1 ; c++ ) {
            System.out.print ("*");
        }
        // Print 2*(r-1) spaces
        for ( int c = 1 ; c< = 2*(r-1) ; c++) {
            System.out.print ("  ");
        }
        // Print N-r+1 stars
        for ( int c = 1 ; c <= N-r+1 ; c++ ) {
            System.out.print ("*");
        }

        System.out.println ();
    }
}
```

N = 4

r	N-r+1	2*(r-1)	N-r+1	print
1	4-1+1=4	2*(1-1)=0	4-1+1=4	********
2	4-2+1=3	2*(2-1)=2	4-2+1=3	***ss***
3	4-3+1=2	2*(3-1)=4	4-3+1=2	**ssss**
4	4-4+1=1	2*(4-1)=6	4-4+1 =1	*ssssss*

Q11> Given N as input, Print the pattern as shown

N = 4

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

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

N = 4

r	left spaces	stars	right spaces
1	3	1	3
2	2	3	2
3	1	5	1
4	0	7	0
	N-r	(2*r)-1	N-r

```
public static void main (....) {
        // Initialise Scanner .....
        int N = scanner.nextInt(); // N=3
        for (int r = 1 ; r <= N ; r++) {
                // Print N-r spaces
                for ( int c = 1 ; c <= N-r ; c++) {
                        System.out.print("  ");
                }
                // Print 2*r - 1
                for (int c = 1 ; c< = 2*r -1 ; c++) {
                        System.out.print("*");
                }
                // Print N-r spaces
        for ( int c = 1 ; c <= N-r    ; c++ ) {
                System.out.print("  ");
        }

        System.out.println();
    }
}
```