

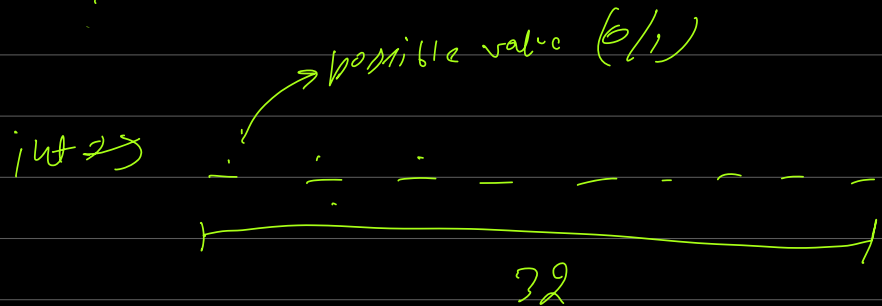
(machine code) $(10101 \dots 11 \dots)$

$$q = 100$$

$$b = \underline{120}$$

$$L = 90$$

$$\max = \cancel{-100}$$



$$\begin{aligned} \max &= \left[\frac{(2^{32} - 1)}{\underline{\underline{\quad}}} \right] \\ \min &= \left[\frac{-2^{32}}{\underline{\underline{\quad}}} \right] \end{aligned}$$

Function

return

Arguments

void

int

✓

x

x

✓

✓

✓

```

      *
    * * *
  * * * * *
* * * * * *

```

$n = 2, 4, 5$

1) 2
 2) 2
 3) 2
 4) 2

```

  *
 * *
* * *
* * * *
* * * * *
* * * * * *

```

```

public static void printTriangle(int sizeOfTriangle) {
    /**
     * nst: number of stars.
     */
    int nst = 1;
    for (int row = 1; row <= sizeOfTriangle; row++) {
        // cst: count of star
        for (int cst = 1; cst <= nst; cst++) {
            System.out.print(" * ");
        }

        System.out.println();
        nst += 2;
    }
}

```

$nst = 1, 3, 5, 7, 9, 11$

```

  *
 * * *
* * * * *
* * * * * *
* * * * * * *

```

row = 5

```

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

N	Space	Star
1	4	1
2	3	3
3	2	5
4	1	7
5	0	9

(row-1)

row = 5

```

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

N	CSP1	hgt	CSP2
	row/2	1	row/2
1	2	1	2
2	1	3	1
3	0	5	0
4	1	3	1
5	2	1	2

```

row 5 → 2 1 2
row 7 → 3 1 3
row 9 → 4 1
  
```

$N = 5$

```
int nst = 1;
int nsp = N / 2;

for (int row = 1; row <= N; row++) {
    // cst: count of space
    for (int csp = 1; csp <= nsp; csp++) {
        System.out.print(s: " ");
    }

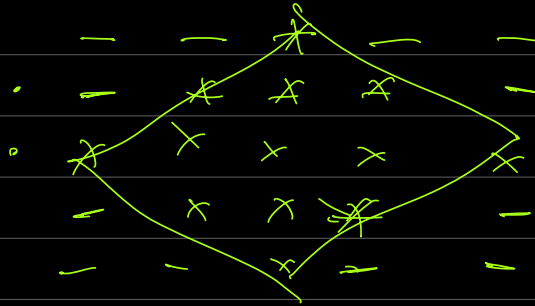
    // cst: count of star
    for (int cst = 1; cst <= nst; cst++) {
        System.out.print(s: "x");
    }

    // cst: count of space
    for (int csp = 1; csp <= nsp; csp++) {
        System.out.print(s: " ");
    }

    System.out.println();

    if (row < N / 2) {
        nst += 2;
        nsp -= 1;
    } else {
        nst -= 2;
        nsp += 1;
    }
}
```

$row = 1, nst = 1, nsp = 2$



hollow diamond

