

```

3 3
1 2 3
4 5 6
7 8 9

```

0 1 2 col: m-1
row: 4

0	1	2	3
1	4	5	6
2	7	8	9

$i = 2 \times 0$
 $j = 2 \times 0$

9 8 7
6 5 4
3 2 1

Sample Output 0

```

9 8 7
6 5 4
3 2 1

```

```

public static void reverseMatrixPrint(int[][] mat) {
    int n = mat.length, m = mat[0].length;
    for (int i = n - 1; i >= 0; i--) {
        for (int j = m - 1; j >= 0; j--) {
            System.out.print(mat[i][j] + " ");
        }
        System.out.println();
    }
}

```

Sample Input 0

6
3 6 9 3 3 9

Sample Output 0

4

Explanation 0

There are 4 lucky pairs (0,3), (0,4), (3,4), (2,5).

3	6	9	3	0	9
0	1	2	3	4	5

$i=0, j=(0,5)$
 $i=1, j=(2,5)$
 $i=2, j=(2,5)$
 $i=3, j=(4,5)$
 $i=4, j=(5,5)$

$(0,1)$, $(0,2)$, $(0,3)$, $(0,4)$, $(0,5)$
 $(1,2)$, $(1,3)$, $(1,4)$
 $(2,3)$, $(2,4)$, $(2,5)$
 $(3,4)$, $(3,5)$
 $(4,5)$

$$h \quad \frac{h}{2} \quad \frac{h}{4} \quad \frac{h}{8} \quad \frac{h}{16} \quad \frac{h}{32} \dots \dots \dots 1, 2, 1$$

$$q_k = q_0 \cdot 2^{k-1}$$

$$1 = h \left(\frac{1}{2}\right)^k$$

$$(2^k = h)$$

$$q_0 = h$$

$$q_k = 1$$

$$r = \left(\frac{1}{2}\right)$$

$$\log_2 2^k = \log_2(h)$$

$$k \log_2 2 = \log_2(h)$$

$$K = \log_2(h)$$

$$N = 10^{12}$$

$$10^9 \text{ operations} = 1s$$

$$10^{12} \text{ operations} = \frac{1}{10^3} \times 10^{12} s$$

$$= 10^9 s$$

$$= \left(\frac{1000}{60}\right) \text{ hrs}$$

$$= 16 \text{ hrs}$$

$$10^9 \text{ ops} = 1s$$

$$10^{12} \text{ ops} = \frac{1}{10^3} \times 10^9$$

$$= \left(\frac{1}{10^3}\right)$$

$$\log_2(N) \approx \log_2 10^{12}$$

$$= 12 \log_2(10)$$

$$= \frac{12 \times 10}{2}$$

$$= 60 \text{ ops}$$

$$(10^9 \text{ ops} = 1s)$$

```

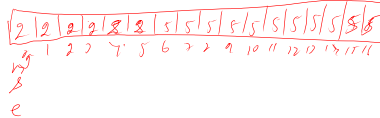
public static void firstLast(int[] arr, int data) {
    int n = arr.length, fo = -1, lo = -1; // fo: first occurrence, lo: last occurrence
    for (int i = 0; i < n; i++) {
        if (arr[i] == data) {
            if (fo == -1)
                fo = i;
            lo = i;
        }
    }
    System.out.println("First Occurrence = " + fo);
    System.out.println("Last Occurrence = " + lo);
}

```

fo = 2, lo = 5

Input:
 • arr[] = {1, 3, 5, 5, 5, 5, 6, 123, 125}
 • x = 5
 Output:
 • First Occurrence = 2
 • Last Occurrence = 5

2



h = 10

```

public static int firstOccurrence(int[] arr, int data) {
    int n = arr.length, si = 0, ei = n - 1;
    while (si <= ei) {
        int mid = (si + ei) / 2;
        if (arr[mid] == data) {
            if (mid - 1 >= 0 && arr[mid - 1] == data) {
                ei = mid - 1;
            } else
                return mid;
        } else if (arr[mid] > data)
            ei = mid - 1;
        else
            si = mid + 1;
    }
    return -1;
}

```

0120
 (1 > 0)