**1st Question**

Given a non-negative integer x, return the square root of x rounded down to the nearest integer. The returned integer should be non-negative as well. You must not use any built-in exponent function or operator.

**Example 1:**

**Input:** x = 4 **Output:** 2 **Explanation:** The square root of 4 is 2, so we return 2.

**Example 2:**

**Input:** x = 8 **Output:** 2 **Explanation:** The square root of 8 is 2.82842..., and since we round it down to the nearest integer, 2 is returned.

**Constraints:**

0 <= x <= 2^31 - 1

**ANS:-**

**def MeraSquareroot(n):**

**if x == 0:**

**return 0**

**a= 1**

**b = n**

**while a <= b:**

**mid = a + (b - a) // 2**

**if mid \* mid > n:**

**b = mid - 1**

**else:**

**a = mid + 1**

**return b**