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

In [1]:

```python
def mySqrt(x):
    if x == 0 or x == 1:
        return x

    guess = x / 2
    tolerance = 0.00001

    while abs(guess * guess - x) > tolerance:
        guess = (guess + x / guess) / 2

    return int(guess)

# Example usage:
x = 4
print(mySqrt(x))  # Output: 2

```

```
2
```

In [2]:

```python
# Example usage:
x = 8
print(mySqrt(x))  # Output: 2
```

```
2
```

In [ ]:

```python

```