Sqrt(x)

Given a non-negative integer `x`, compute and return *the square root of* `x`.

Since the return type is an integer, the decimal digits are **truncated**, and only **the integer part** of the result is returned.

Note: You are not allowed to use any built-in exponent function or operator, such as `pow(x, 0.5)` or `x ** 0.5`.

Example 1:

```
Input: x = 4

Output: 2
```

Example 2:

```
Input: x = 8

Output: 2

Explanation: The square root of 8 is 2.82842..., and since the decimal part is truncated, 2 is returned.
```

Constraints:

- $0 <= x <= 2^{31} - 1$

Program :

```
class Solution {
   public int mySqrt(int x) {
      if(x==0)
      {
         return(x);
      }
      long l=1;long r =x;
      long mid; long res =0;
      while(l<=r)
```

```
        {
            mid = l + (r-l)/2;
            if(mid*mid < x)
            {
                res =mid;
                l = mid+1;

            }else if(mid*mid > x){ r = mid -1;}
            else{ return (int) mid;}
        }
        return (int) res;

    }
}
```

Output :

Accepted

Runtime: 0 ms

Your input

```
4
```

Output

```
2
```

Diff

Expected

```
2
```