

输入一个非负整数 n ，请你计算阶乘 $n!$ 的结果末尾有几个 0。

分析： 结果的末尾的 0 从哪里来的？

1. 两个数相乘结果末尾有 0，一定是因为两个数中有因子 2 和 5，因为 $10 = 2 \times 5$ 。

问题转化为： $n!$ 最多可以分解出多少个因子 2 和 5？ 主要取决于能分解出几个因子 5，因为每个偶数都能分解出因子 2，因子 2 肯定比因子 5 多得多。问题转化为： $n!$ 最多可以分解出多少个因子 5？ 只需计算 n 中有多少个 5 的倍数， 5^n 的倍数，求他们之和即可。

```
class Solution {
public:
    int trailingZeroes(int n) {
        int res = 0;
        long divisor = 5;
        while (divisor <= n) {
            res += n / divisor;
            divisor *= 5;
        }

        return res;
    }
};
```

输入一个非负整数 K ，请你计算有多少个 n ，满足 $n!$ 的结果末尾恰好有 K 个 0。

搜索有多少个 n 满足 $\text{trailingZeroes}(n) == K$ ，其实就是在问，满足条件的 n 最小是多少，最大是多少，最大值和最小值一减，就可以算出来有多少个 n 满足条件了。

```
class Solution {
public:
    long preimageSizeFZF(long k) {
        return right_bound(k) - left_bound(k) + 1;
    }

private:
    // 查找左边界
    long left_bound(long k) {
        long left = 0, right = LONG_MAX;

        while (left < right) {
            long mid = left + (right - left) / 2;
            long tmp = trailingZeroes(mid);
            if (k == tmp) {
                right = mid;
            }
        }
    }
};
```

```
        } else if (tmp < k) {
            left = mid + 1;
        } else {
            right = mid;
        }
    }

    return left;
}

// 查找右边界
long right_bound(long k) {
    long left = 0, right = LONG_MAX;

    while (left < right) {
        long mid = left + (right - left) / 2;
        long tmp = trailingZeroes(mid);
        if (k == tmp) {
            left = mid + 1;
        } else if (tmp < k) {
            left = mid + 1;
        } else {
            right = mid;
        }
    }

    return left - 1;
}

long trailingZeroes(long n) {
    // 计算n的阶乘中0的个数
    long res = 0;

    for (long d = n; d / 5 > 0; d = d / 5) {
        res += d / 5;
    }

    return res;
}
};
```