题目：

Write a program to find the n-th ugly number.

Ugly numbers are positive numbers whose prime factors only include 2, 3, 5. For example, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12 is the sequence of the first 10 ugly numbers.

Note that 1 is typically treated as an ugly number, and *n* **does not exceed 1690**.

Hint:

1. The naive approach is to call isUgly for every number until you reach the n^{th} one. Most numbers are *not* ugly. Try to focus your effort on generating only the ugly ones.
2. An ugly number must be multiplied by either 2, 3, or 5 from a smaller ugly number.
3. The key is how to maintain the order of the ugly numbers. Try a similar approach of merging from three sorted lists: L_1, L_2, and L_3.
4. Assume you have U_k, the k^{th} ugly number. Then U_{k+1} must be Min($L_1 * 2$, $L_2 * 3$, $L_3 * 5$).

```
class Solution {
public:
    int nthUglyNumber(int n) {
        int var_2 = 2, var_3 = 3, var_5 = 5;
```

```
        int index_2 = 0, index_3 = 0, index_5 = 0;
        std::vector<int> dp(n, 0);
        dp[0] = 1;
        for (int i = 1; i < n; ++i){
            int min = std::min(std::min(var_2, var_3), var_5);
            if (min == var_2){
                dp[i] = var_2;
                var_2 = dp[++index_2] * 2;
            }
            if (min == var_3){
                dp[i] = var_3;
                var_3 = dp[++index_3] * 3;
            }
            if (min == var_5){
                dp[i] = var_5;
                var_5 = dp[++index_5] * 5;
            }
        }
        return dp[n - 1];
    }
};
```