

The problem is to determine if a number is a "Limited Factor Number," meaning all its prime factors are limited to 2, 3, and 5. Here's a Python solution for the task:

```
def is_limited_factor_number(x):
    if x <= 0:
        return False # Only positive integers are valid
    if x == 1:
        return True # 1 is considered a limited factor number

    # Divide the number by 2, 3, and 5 as much as possible
    for factor in [2, 3, 5]:
        while x % factor == 0:
            x //= factor

    # If x becomes 1, it means all factors were 2, 3, or 5
    return x == 1

# Example usage:
print(is_limited_factor_number(6)) # Output: True
print(is_limited_factor_number(1)) # Output: True
print(is_limited_factor_number(14)) # Output: False
```

Explanation of the Code:

1. Base Cases:
 - If $x \leq 0$, return False because only positive integers are allowed.
 - If $x = 1$, return True because 1 has no prime factors.
2. Factor Reduction:
 - For each prime factor (2, 3, 5), repeatedly divide x by the factor until it is no longer divisible.
3. Final Check:
 - If x becomes 1 after removing all factors of 2, 3, and 5, then x is a limited factor number.
 - Otherwise, return False.

Complexity:

- Time Complexity: $O(\log x)$, where x is the input number, due to repeated division.
- Space Complexity: $O(1)$, as no extra space is used.

Let me know if you need further clarification!