a. The procedure `p` is applied 6 times. It is applied until the absolute value of `angle` is less than 0.1. For each division, an additional `p` call encloses `angle`, and it takes 5 divisions for 12.15 to be reduced to under 0.1. These are the approximate values of `angle` after each division:

```
(12.15, 4.05, 1.35, 0.45, 0.15, 0.05).
```

b. To calculate the order of growth in space, we must first recognize that the process is *linear recursive*, meaning that it grows a chain of deferred operations (`p` calls). The interpreter must keep track of these deferred `p` calls in memory space. Therefore, understanding the number of `p` calls for a given `a` allows us to calculate the order of growth for space. The number of `p` calls grows once for every increase in the absolute value of `a` such that an additional division by 3 must occur to reduce the absolute value of `a` to be less than 0.1. Starting from an `a` of 3^{-3}, which requires no divisions (requiring only 1 `p` call), `a` must be multiplied by 3 to increase the number of divisions by 1.

a	0.037	0.111	0.333	1	3	9	27	81	243	729	2187	6561
Number of Calls	1	2	3	4	5	6	7	8	9	10	11	12

This yields a formula of $R(n) = \log_3(n) + 4$,where n is the initial absolute value of `a` and $R(n)$ is the maximum number of `p` calls that must be kept in memory.

The order of growth in the number of steps is found by understanding that the number of steps for each `p` call is the same regardless of the value of `x` in `(p x)`. Therefore, the total number of steps, $T(n)$, will be $k \cdot R(n)$, where k is a constant representing the number of steps in a `p` call.

Therefore, the order of growth in space and number of steps (as a function of a), used by the process generated by the `sine` procedure when `(sine a)` is evaluated is $\Theta(\log a)$.