lazy evaluation, or call-by-need, is an evaluation strategy which delays the evaluation of an expression until its value is needed, and which also avoids repeated evaluations .

in this program, we have 2 parts,

in the “eager evaluation” part,

the program first converts all the yield values from the function “generate\_values” into a list.  
This is done eagerly because list() forces the evaluation of all items at once, and immediately.  
and then the values are passed to “square”, the list of generated values already exists in memory.

In the “lazy evaluation” part,

The program uses list comprehension that iterates over “generate\_values”,

And because of that, the values are generated one at a time, and then immediately squared.

Because of that only one of the values exists in memory each time.

It is a lazy evaluation since values are calculated only when necessary and are not stored unnecessarily .

The lazy evaluation in that case uses less memory, its more efficient, and because only one value is processed at the same time, it can potentially handle infinite sequences.