Question #3: Give example for optimization used in mini compiler?

Ans: **Example of Optimization in the Mini Compiler**

The **optimization** function in the mini compiler aims to simplify expressions by evaluating constant expressions at compile time, reducing unnecessary operations. This can improve performance by eliminating redundant calculations.

For example, in an arithmetic expression like "3 + 5", the compiler can optimize it to just "8" instead of keeping it as a runtime operation.

Consider the expression: "3 + (4 \* 5) \* 7"

1. **Initial Tokens**: After lexical analysis, the tokens are:

["NUMBER: 3", "PLUS: +", "LPAREN: (", "NUMBER: 4", "TIMES: \*", "NUMBER: 5", "RPAREN: )", "TIMES: \*", "NUMBER: 7"]

**Optimization Process**:

* The optimizer looks for expressions where both operands are numbers, like 4 \* 5 or 3 + 5.
* It evaluates 4 \* 5 to 20 and replaces it in the token list.
* The tokens are now:

["NUMBER: 3", "PLUS: +", "NUMBER: 20", "TIMES: \*", "NUMBER: 7"]

**Final Optimized Tokens**: The final optimized tokens are:  
 ["NUMBER: 3", "PLUS: +", "NUMBER: 140"]

**Final Expression**: The expression is now simplified to:

3 + 140  
143