Name:	**Mubashir Ali**
Reg #	**FA21-BCS-009**
Course Name:	**Compiler Construction**
Submitted To:	**Syed Bilal Haider Bukhari**
Date:	**03 Jan 2025**

Lab Terminal

Question 3: Give examples of code optimizations you used in your mini compiler.
Answer: In a Mini C Compiler, several code optimization techniques can be implemented to enhance performance and efficiency. Here are five common optimizations:

1. **Constant Folding:**
    - **Description:** Compile-time evaluation of constant expressions to reduce runtime computations.
    - **Example:** Transforming int x = 2 * 3; directly into int x = 6;.
2. **Dead Code Elimination:**
    - **Description:** Removing code segments that do not affect the program's output, thereby reducing code size and improving performance.
    - **Example:** Eliminating an unused variable assignment like int x = 5; when x is never utilized.
3. **Loop Unrolling:**
    - **Description:** Expanding loop iterations to decrease the overhead of loop control, enhancing execution speed.
    - **Example:** Converting a loop such as for (int i = 0; i < 4; i++) { a[i] = i; } into a[0] = 0; a[1] = 1; a[2] = 2; a[3] = 3;.
4. **Common Subexpression Elimination:**
    - **Description:** Identifying and computing repeated expressions once, storing the result for subsequent uses to save computation time.
    - **Example:** Replacing multiple instances of a * b in a function with a single computed value, assuming a and b remain unchanged.
5. **Inline Expansion:**
    - **Description:** Replacing function calls with the function's body to eliminate call overhead, especially for small, frequently called functions.
    - **Example:** Substituting a call to a simple function like int square(int x) { return x * x; } directly with x * x in the code.

Implementing these optimizations can significantly improve the efficiency of the compiled code.