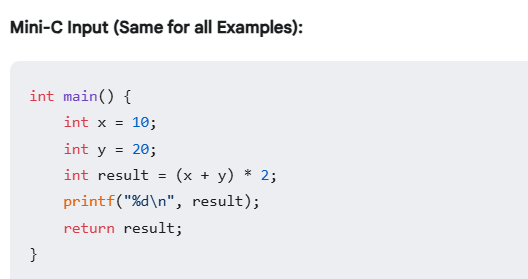
Q3: Give Optimizations used in your miniC compiler?

Ans:

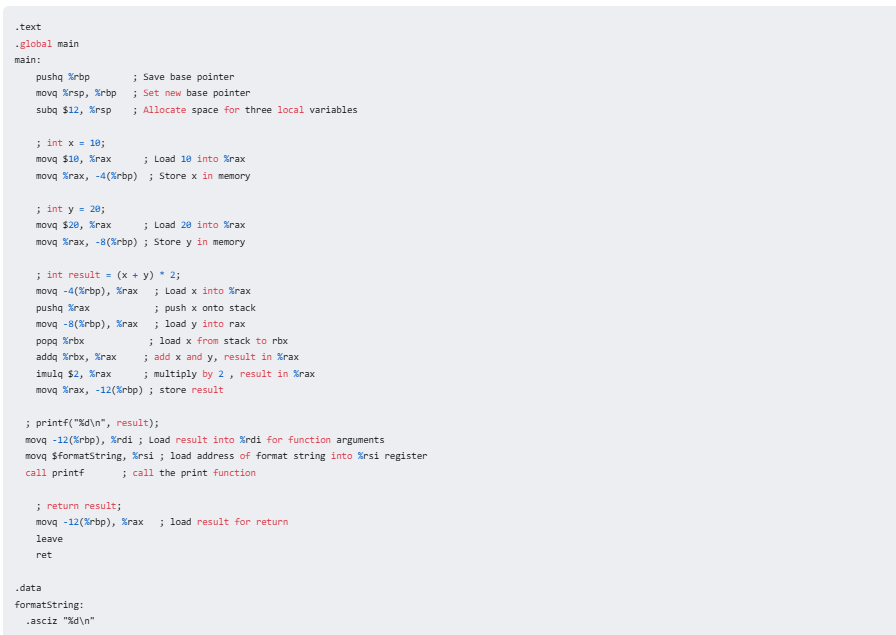
**Mini-C Input (Same for all Examples):**

Input:



**1. Unoptimized Assembly (output\_unoptimized.s)**

This is the baseline, with no optimizations:



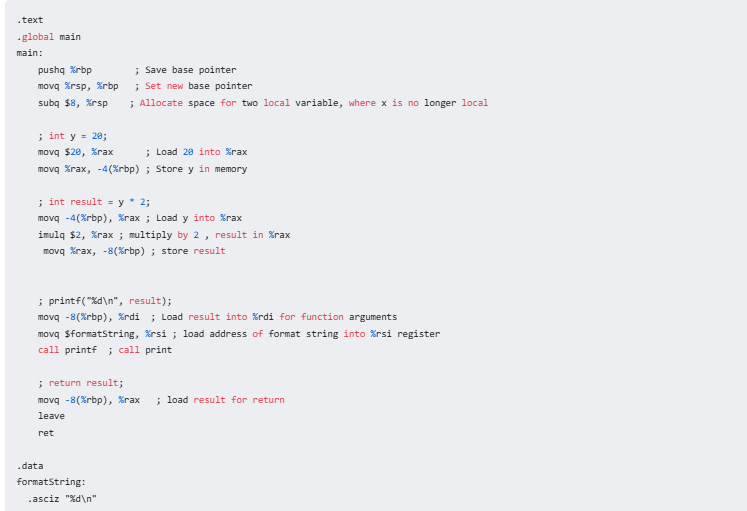
**2. Assembly with Constant Folding (output\_constant\_folded.s)**



**Explanation of Changes in output\_constant\_folded.s:**

* **Calculation Replaced:** The code to calculate (x + y) \* 2 is replaced with movq $60, %rax. The compiler did the calculation at compile time.
* **Memory access:** The result is stored in memory, at the location of the variable result.

**3. Assembly with Dead Code Elimination (output\_deadcode\_eliminated.s):**



* **Eliminated x Instructions:** The code for int x = 10; is completely gone, since x is never used in this program. The instructions movq $10, %rax and movq %rax, -4(%rbp) are removed. This is because the variable is never used.
* **Reduced Stack allocation:** Since the variable x is not being stored to the stack anymore, then the number of bytes that are reserved are now $8 instead of $12.