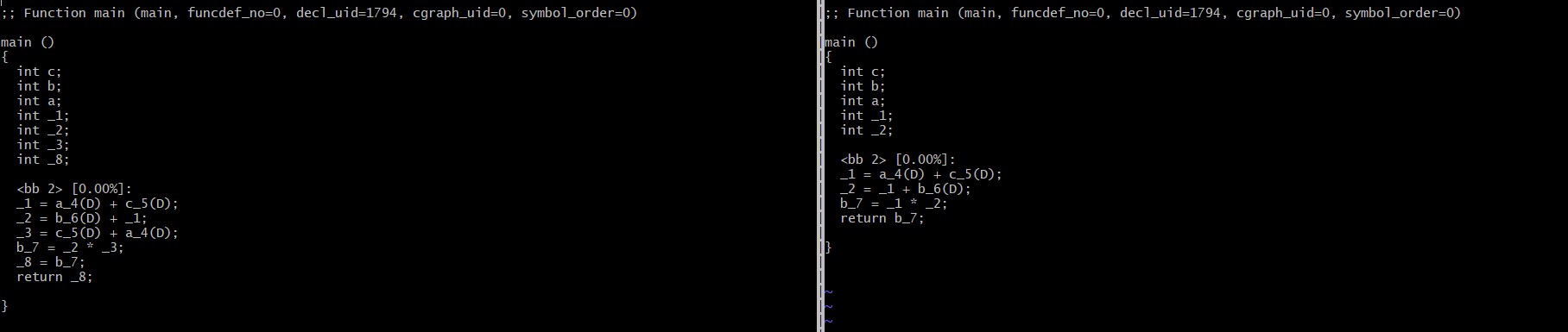
# Output



# Input

int main()

{

int a, b, c;

b = (a + c + b) \* (c + a);

return b;

}

# Answers

1]

The original code showed two computations, but the FRE pass was able to reuse the temporary variable which stored the result of the initial computation of (a + c). Thus, one line of execution was saved, and so was the use of an extra variable.

2]

Check the file B-4.c.\*.optimized. This is the last GIMPLE intra-procedural pass. If it is same as the output of FRE, then no more optimizations have occurred.

3] "return b" was required to make the computation of variable `b' meaningful. If function ‘main’ is going to "return 0" and there is no other output, GCC will realise that the entire computation of

variable `b' was dead code and will eliminate its dependencies one by one until the resulting program reduces to:

int main()

{

return 0;

}