## ****Questions****

Try code the matrix (eg. 8x8) multiplication in a straight forward algorithm.

Compile it with various optimizations.

See if the compiler is able to perform loop interchange.

Try this simple code. See if the compiler is able to perform any optimization.

Please provide your analysis.

You can see whole project here <https://github.com/chanakorn-github/HPA-loop-tranformation>

## Analysis

Using compiler (x86-64 gcc 12.2) from <https://godbolt.org/>

**In O0**

<https://github.com/chanakorn-github/HPA-loop-tranformation/blob/main/O0.s>

There does not appear to be any loop transformation present in this code.

**In O1**

<https://github.com/chanakorn-github/HPA-loop-tranformation/blob/main/O1.s>

there are loops in the assembly code you provided. The outer loop starts at label **.L3** and iterates for **rax** times, where **rax** is calculated earlier in the code. The inner loop starts at label **.L6** and iterates for **r9** times, where **r9** is also calculated earlier in the code.

**In O2**

<https://github.com/chanakorn-github/HPA-loop-tranformation/blob/main/O2.s>

I don't see a straightforward loop transformation to improve performance.

**In term of time**

<https://github.com/chanakorn-github/HPA-loop-tranformation/blob/main/time.cpp>   
A screenshot of a computer

Description automatically generated with medium confidence

There is slightly different which the O1 has the best performance. However, it's worth noting that the specific optimizations that result in better performance can vary depending on the code, hardware, and other factors.

**this is my computer**

|  |
| --- |
| Device name LAPTOP-6M7OQAR6  Processor 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz 2.69 GHz  Installed RAM 16.0 GB (15.7 GB usable)  Device ID 754EA662-2411-4CD6-A13D-0D65CF4BDB87  Product ID 00327-35942-57465-AAOEM  System type 64-bit operating system, x64-based processor  Pen and touch No pen or touch input is available for this display |