CS 590: Topics in Computer Science
Assignment 07: x86_64 Assembly Instructions

This assignment is about understanding the x86_64 microprocessor architecture instructions through the assembly language automatically generated by the GNU C Compiler.
Write a C program using a global variable of type double, a simple program, and compile it in the Linux server (or any Linux machine) using:

```
gcc -S -O0 global01.c
gcc -g -O0 global01.c -o global01
```

Compare the `global01.s` with the code that the GDB disassembler returns. Run the program step by step and explain how the instructions work to fulfill the goals of the program.

```
gdb global01
(gdb) break main
(gdb) run
(gdb) disassemble
```

More about the GDB in the following link:
https://sourceware.org/gdb/current/onlinedocs/gdb/
You can also check the GDB manual pages through the terminal typing:

```
man gdb
```

Write another program using an array and an “if” **following the same procedure** as in the previous one.

Repeat all what you have done but now using the optimization flag set to “-O1”.
About the optimization flags:

> **-O0**
> Do not optimize. This is the default.
>
> **-O**
> **-O1**
> Optimize. Optimizing compilation takes somewhat more time, and a lot more memory for a large function.
> With '-O', the compiler tries to reduce code size and execution time, without performing any optimizations that take a great deal of compilation time.
> '-O' turns on the following optimization flags: `-fdefer-pop -fmerge-constants -fthread-jumps -floop-optimize -fcrossjumping -fif-conversion -fif-conversion2 -fdelayed-branch -fguess-branch-probability -fcprop-registers`
> '-O' also turns on '`-fomit-frame-pointer`' on machines where doing so does not interfere with debugging.
>
> **-O2**
> Optimize even more. GCC performs nearly all supported optimizations that do not involve a space-speed tradeoff. The compiler does not perform loop unrolling or function inlining when you specify '-O2'. As compared to '-O', this option increases both compilation time and the performance of the generated code.

'-O2' turns on all optimization flags specified by '-O'. It also turns on the following optimization flags:
`-fforce-mem -foptimize-sibling-calls -fstrength-reduce -fcse-follow-jumps -fcse-skip-blocks -frerun-cse-after-loop -frerun-loop-opt -fgcse -fgcse-lm -fgcse-sm -fdelete-null-pointer-checks -fexpensive-optimizations -fregmove -fschedule-insns -fschedule-insns2 -fsched-interblock -fsched-spec -fcaller-saves -fpeephole2 -freorder-blocks -freorder-functions -fstrict-aliasing -falign-functions -falign-jumps -falign-loops -falign-labels`
Please note the warning under '-fgcse' about invoking '-O2' on programs that use computed gotos.

-O3
Optimize yet more. '-O3' turns on all optimizations specified by '-O2' and also turns on the '`-finline-functions`' and '`-frename-registers`' options.

-Os
Optimize for size. '-Os' enables all '-O2' optimizations that do not typically increase code size. It also performs further optimizations designed to reduce code size.

'-Os' disables the following optimization flags: `-falign-functions -falign-jumps -falign-loops -falign-labels -freorder-blocks -fprefetch-loop-arrays`

If you use multiple '-O' options, with or without level numbers, the last such option is the one that is effective.

Questions/Tasks:

1. Explain about the Microarchitectural Data Sampling (MDS) vulnerability. What it is, why it works and how to mitigate it.
2. Explain what a Microcode is and its relation with the hardware. Also explain why keeping it up to date is an important security measure.

SUBMISSION
Write your report including screenshots of your program running and explaining how it works. Include comments in your code too and submit your program source code files (.c and .asm). Submit your report as a DOCX or PDF document through Western Online. Include in the report the answers to the questions writing the corresponding numbers and questions (or at least the numbers) in **bold** and in the proper order before every answer. At the end include a conclusion (properly labeled as that) explaining the importance of all this.

Name your report using the following format:
`YourLastName_YourFirstName_CS590_Lab07.pdf`

---///