This submission did not make use of a machine that is running Linux. Rather, this submission was done on a Windows 11 machine using an MSYS2 MinGW 64-bit Shell in order to simulate multithreading via OpenMP.

Additionally, I used `time`, which is shell-provided, in order to get the machine's execution time of each task.

These commands were used to compile the C files for each task:

```shell
gcc -O2 task1.c -o task1
gcc -O2 -mavx2 task2.c -o task2
gcc -O2 task3.c -o task3 -fopenmp
gcc -O2 -mavx2 task4.c -o task4 -fopenmp
```

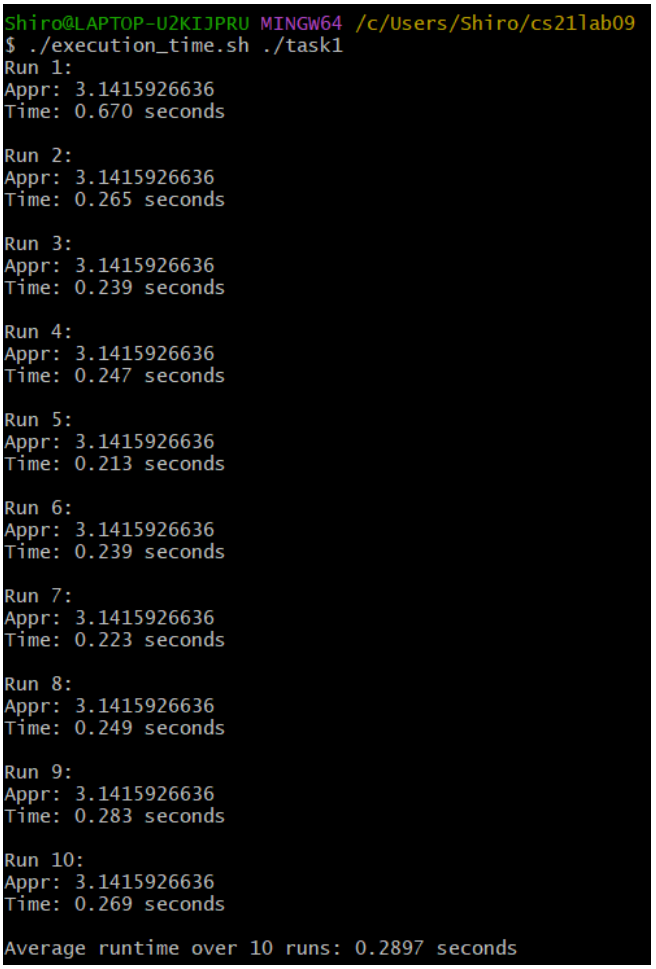
I made use of the shell-provided `time` instruction in order to get execution time for running a task. I then made a shell script `execution_time.sh` which runs a task 10 times, and then gets the average run time of the runs. For example, for running task 1:

```shell
./execution_time.sh ./task1
```

Execution Times

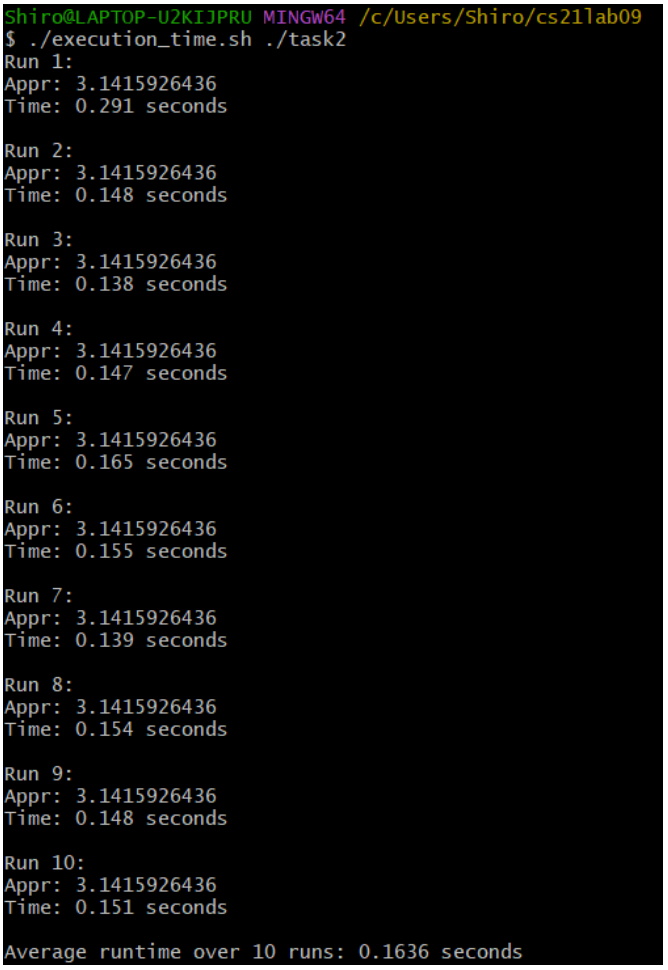
Task 1 - Basic Implementation

```shell
gcc -O2 task1.c -o task1
./execution_time.sh ./task1
```



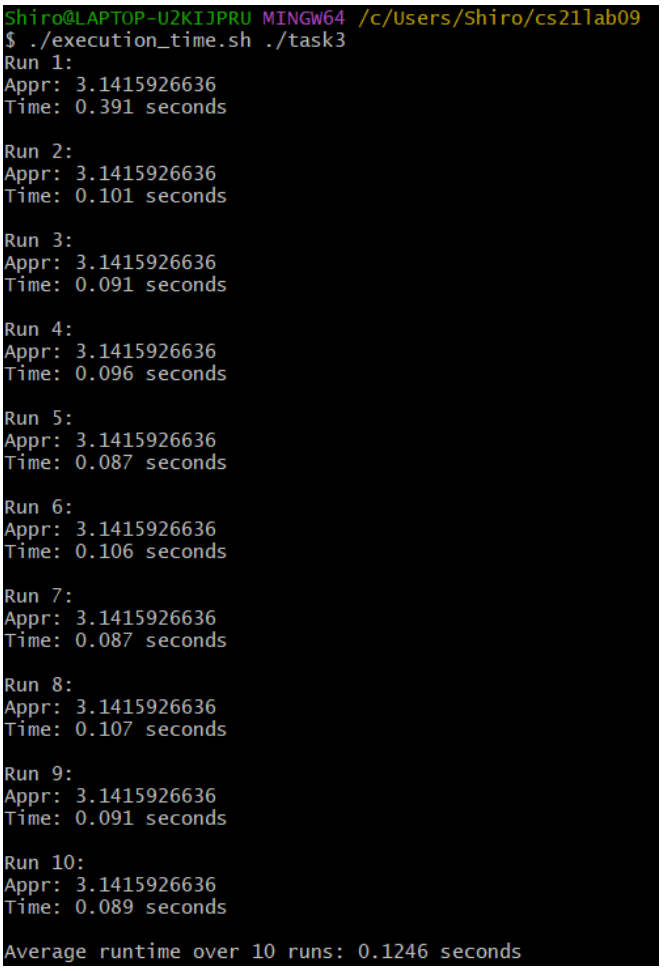
Task 2- Vectorized Implementation

```shell
gcc -O2 -mavx2 task2.c -o task2
./execution_time.sh ./task2
```



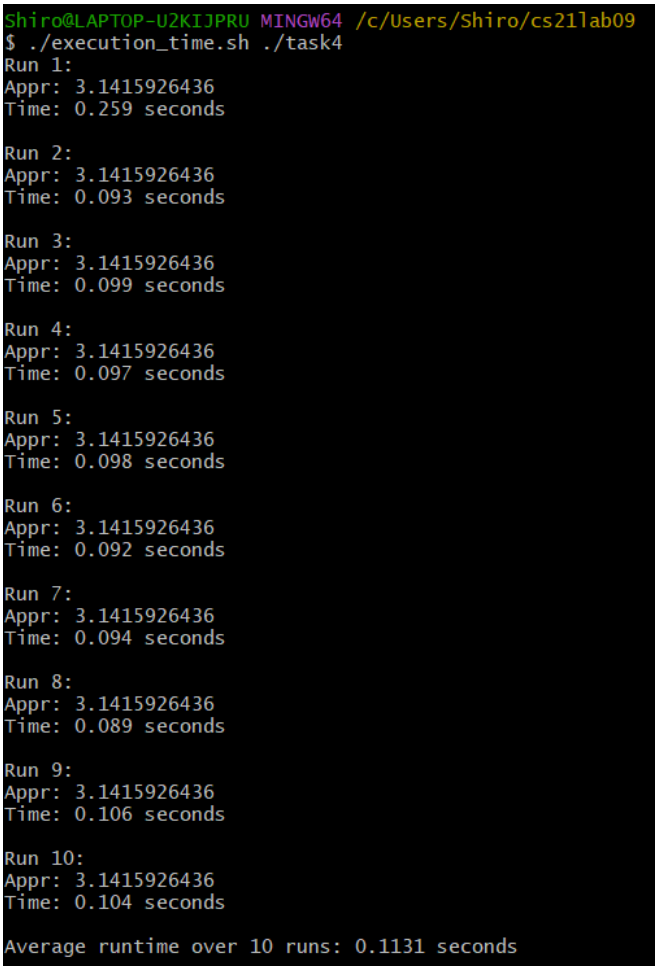
Task 3 - Multicore Implementation

```shell
gcc -O2 task3.c -o task3 -fopenmp
./execution_time.sh ./task3
```



Task 4 - Hybrid Implementation

```shell
gcc -O2 -mavx2 task4.c -o task4 -fopenmp
./execution_time.sh ./task4
```



Speedup Computations

Say you want to compute the speedup from Implementation A to Implementation B, the formula to be used for the Speedup is:

$$Speedup_{A,B} = \frac{AvgRuntime_A}{AvgRuntime_B}$$

$$Speedup_{1,2} = \frac{AvgRuntime_1}{AvgRuntime_2} = \frac{0.2897}{0.1636} = 1.7708\ times\ faster$$

$$Speedup_{1,3} = \frac{AvgRuntime_1}{AvgRuntime_3} = \frac{0.2897}{0.1246} = 2.3250\ times\ faster$$

$$Speedup_{1,4} = \frac{AvgRuntime_1}{AvgRuntime_4} = \frac{0.2897}{0.1131} = 2.5615\ times\ faster$$

$$Speedup_{2,4} = \frac{AvgRuntime_2}{AvgRuntime_4} = \frac{0.1636}{0.1131} = 1.4465\ times\ faster$$

$$Speedup_{3,4} = \frac{AvgRuntime_3}{AvgRuntime_4} = \frac{0.1246}{0.1131} = 1.1017\ times\ faster$$