COA Assignment - 1

Pranav Patil
22110199

Q1)

a. Time taken by recursion: 109.376 seconds
b. Time taken by loop: 3e-05 seconds
c. Time taken by recursion with memoization: 3.1e-05 seconds
d. Time taken by loop with memoization: 3.4e-05 seconds

To calculate the speedup ratios, you need to compare the time taken by each method to the time taken by recursion, which is your baseline. The formula for speedup ratio is:

Speedup Ratio=Time taken by method/Time taken by baseline

Here's the calculation for each method:

a. **Recursion vs. Loop**:
   Speedup Ratio = 109.376 / 3e-05 = 3,645,866

b. **Recursion vs. Recursion with Memoization**:
   Speedup Ratio = 109.376 / 3.1e-05 = 3,527,451

c. **Recursion vs. Loop with Memoization**:
   Speedup Ratio = 109.376 / 3.4e-05 = 3,217,529

So the speedup ratios compared to the recursion baseline are approximately:

- **Loop**: 3,645,866
- **Recursion with Memoization**: 3,527,451
- **Loop with Memoization**: 3,217,529

Q2a)

System/CPU time as N varies for C++ integer operations

Time taken for Integer matrix multiplication of size **64x64**: 0.005388 seconds
./a.out 0.01s user 0.00s system 3% cpu 0.349 total

Time taken for Integer matrix multiplication of size **128x128**: 0.032153 seconds
./a.out 0.04s user 0.00s system 9% cpu 0.420 total

Time taken for Integer matrix multiplication of size **256x256**: 0.136549 seconds
./a.out 0.14s user 0.00s system 27% cpu 0.534 total

Time taken for Integer matrix multiplication of size **512x512**: 0.752938 seconds
./a.out 0.77s user 0.00s system 43% cpu 1.772 total

Time taken for Integer matrix multiplication of size **1024x1024**: 5.89173 seconds
./a.out 5.93s user 0.01s system 82% cpu 7.173 total

N	System Time	CPU Time
64	0.349	0.01
128	0.420	0.04
256	0.534	0.14
512	1.772	0.77
1024	7.173	5.94

Table 1: System/CPU time as N varies for C++ integer operations

System/CPU time as N varies for C++ float operations

Time taken for Double matrix multiplication of size **64x64:** 0.005706 seconds
./a.out 0.01s user 0.00s system 3% cpu 0.348 total

Time taken for Double matrix multiplication of size **128x128:** 0.026751 seconds
./a.out 0.03s user 0.00s system 3% cpu 0.980 total

Time taken for Double matrix multiplication of size **256x256:** 0.132606 seconds
./a.out 0.14s user 0.00s system 11% cpu 1.266 total

Time taken for Double matrix multiplication of size **512x512:** 0.785699 seconds
./a.out 0.80s user 0.01s system 66% cpu 1.201 total

Time taken for Double matrix multiplication of size **1024x1024:** 6.91531 seconds
./a.out 6.96s user 0.02s system 95% cpu 7.326 total

N	System Time	CPU Time
64	0.348	0.01
128	0.980	0.03
256	1.266	0.14
512	1.201	0.81
1024	7.326	6.98

Table 2: System/CPU time as N varies for C++ float operations

System/CPU time as N varies for python integer operations

Time taken for Integer matrix multiplication of size 64x64: 0.107969 seconds
python3 q2a.py 1.24s user 0.55s system 620% cpu 0.288 total

Time taken for Integer matrix multiplication of size 128x128: 0.635660 seconds
python3 q2a.py 2.14s user 0.54s system 352% cpu 0.760 total

Time taken for Integer matrix multiplication of size 252x252: 4.407973 seconds
python3 q2a.py 5.71s user 0.77s system 143% cpu 4.510 total

Time taken for Integer matrix multiplication of size 512x512: 37.624068 seconds
python3 q2a.py 38.68s user 0.96s system 105% cpu 37.738 total

Time taken for Integer matrix multiplication of size 1024x1024: 309.580205 seconds
python3 q2a.py 309.98s user 0.74s system 100% cpu 5:09.71 total

N	System Time	CPU Time
64	0.288	1.79
128	0.760	2.68
256	4.510	6.48
512	37.738	39.64
1024	309.71	310.72

Table 3: System/CPU time as N varies for python integer operations

System/CPU time as N varies for python float operations

Time taken for Double matrix multiplication of size 64x64: 0.110574 seconds
python3 q2a.py 1.43s user 0.03s system 619% cpu 0.236 total

Time taken for Double matrix multiplication of size 128x128: 0.670357 seconds
python3 q2a.py 2.46s user 0.33s system 351% cpu 0.795 total

Time taken for Double matrix multiplication of size 256x256: 4.802179 seconds
python3 q2a.py 5.95s user 0.92s system 139% cpu 4.914 total

Time taken for Double matrix multiplication of size 512x512: 39.336604 seconds
python3 q2a.py 41.30s user 0.03s system 104% cpu 39.461 total

Time taken for Double matrix multiplication of size 1024x1024: 738.520173 seconds
python3 q2a.py 321.97s user 0.34s system 43% cpu 12:18.64 total

N	System Time	CPU Time
64	0.236	1.46
128	0.795	2.79
256	4.914	6.87
512	39.461	41.33
1024	738.64	322.31

Table 4: System/CPU time as N varies for python float operations

Q2B)

N	Meat Time	Total Time	Percentage
64	0.005388	0.349	1.518
128	0.032153	0.420	7.619
256	0.136549	0.534	25.468
512	0.752938	1.772	42.437
1024	5.89173	7.173	82.113

Table 5: Meat/Total time as N varies for C++ integer operations

N	**Meat Time**	**Total Time**	**Percentage**
64	0.005706	0.348	1.436
128	0.026751	0.980	2.042
256	0.132606	1.266	10.268
512	0.785699	1.201	65.362
1024	6.91531	7.326	94.321

Table 6: Meat/Total time as N varies for C++ float operations

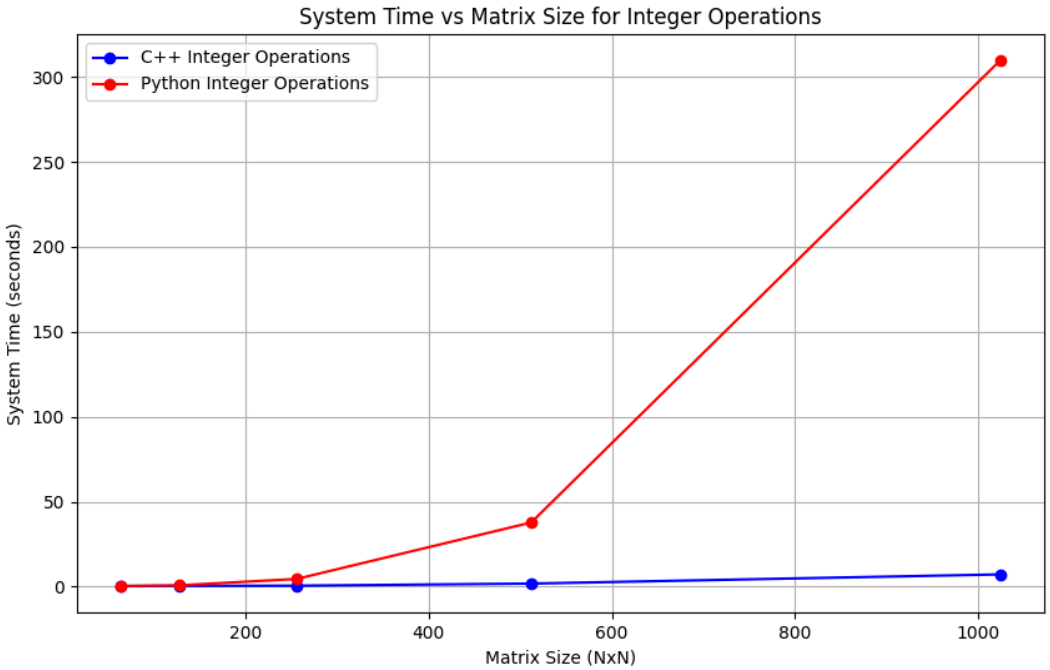
N	Meat Time	Total Time	Percentage
64	0.107969	0.288	37.152
128	0.635660	0.760	83.552
256	4.407973	4.510	97.716
512	37.624068	37.738	99.697
1024	309.580205	309.71	99.958

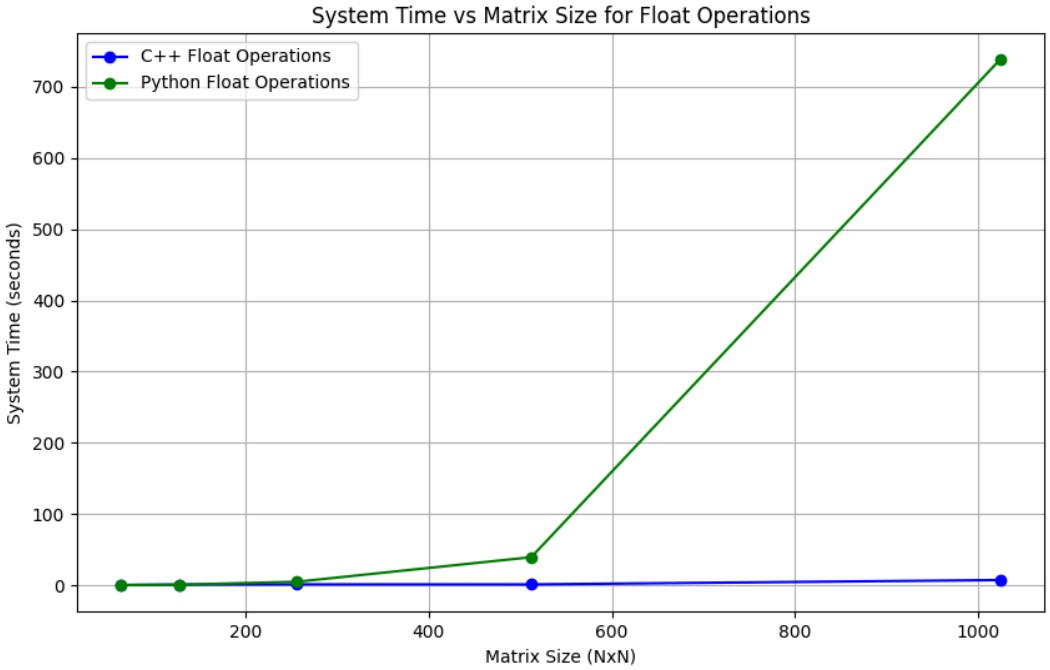
Table 7: Meat/Total time as N varies for python integer operations

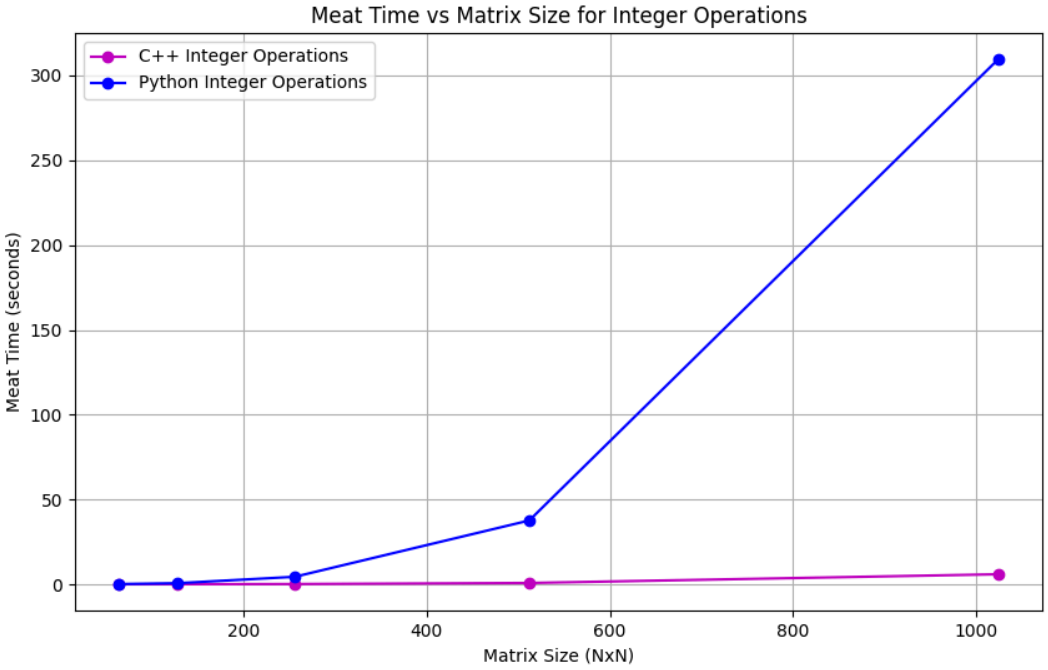
N	Meat Time	Total Time	Percentage
64	0.110574	0.236	46.610
128	0.670357	0.795	84.276
256	4.802179	4.914	97.720
512	37.624068	39.461	95.344
1024	738.520173	738.64	99.983

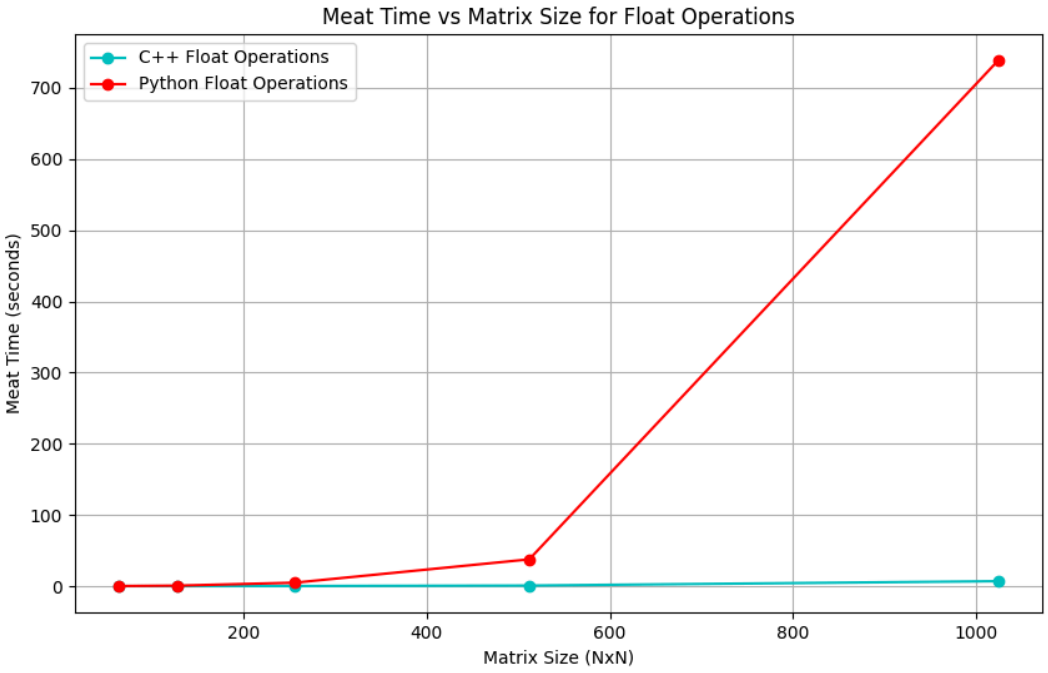
Table 8: Meat/Total time as N varies for python float operations

Q2C)


System Time vs Matrix Size for Integer Operations
C++ Integer Operations
Python Integer Operations
System Time (seconds)
Matrix Size (NxN)
0
50
100
150
200
250
300
200
400
600
800
1000


System Time vs Matrix Size for Float Operations
C++ Float Operations
Python Float Operations
System Time (seconds)
Matrix Size (NxN)
0
100
200
300
400
500
600
700
200
400
600
800
1000


Meat Time vs Matrix Size for Integer Operations
C++ Integer Operations
Python Integer Operations
Meat Time (seconds)
Matrix Size (NxN)
0
50
100
150
200
250
300
200
400
600
800
1000


Meat Time vs Matrix Size for Float Operations
C++ Float Operations
Python Float Operations
Meat Time (seconds)
Matrix Size (NxN)
0
100
200
300
400
500
600
700
200
400
600
800
1000

Conclusion:

1. Reasons Why C++ Performs Better Than Python:
   C++ is a compiled language, meaning that code is converted into machine code before execution. This leads to faster execution times as the code is optimized for the specific architecture.
   Python is an interpreted language, which means the code is executed line-by-line by the interpreter, introducing additional overhead.

2. As N increases, both system and execution times grow for matrix multiplication in both languages. However, the increase is much more pronounced in Python compared to C++.