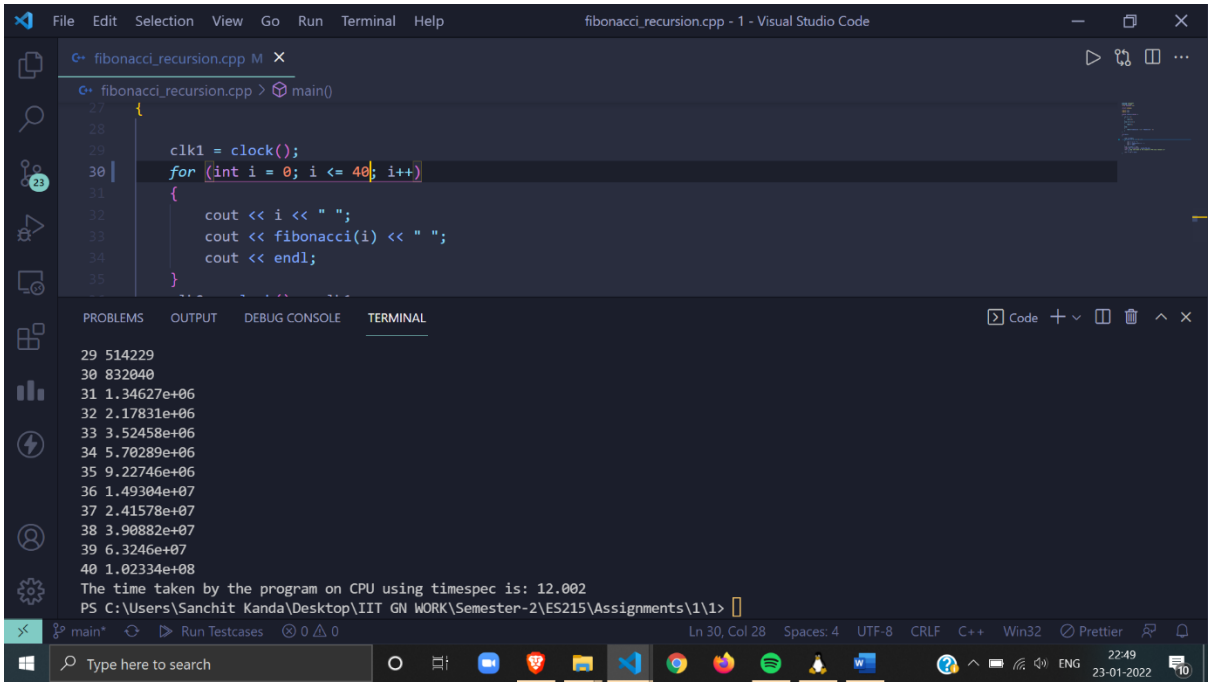
ES215: Computer Organisation and Architecture

[Github Repo Link](https://github.com/Madhav-Kanda/ES215_Assignment_1)

https://github.com/Madhav-Kanda/ES215_Assignment_1

Q1.

a) Since we cannot run the program for 100 Fibonacci numbers as it would take a very long time, therefore, using the time calculated for the first 40 Fibonacci numbers we calculate the time for
Time taken for first 40 Fibonacci =**12.002 sec**
Formula to calculate time taken for 100 Fibonacci = ((1.168)^(60)))*(12.002) sec = **133,605.45 sec**
Time taken using recursion: **133,605.45 sec**



The above image shows the time taken by the Recursion to calculate first 40 Fibonacci numbers.

b) Time taken using loop for 5 observations are as follows:

```
99 2.18923e+20
100 3.54225e+20
The time taken by the program on CPU using loop is: 0.127
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop is: 0.109
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop is: 0.113
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop is: 0.1
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop is: 0.08
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

The values in the sorted order are as follows:

0.08,0.1,0.109,0.113,0.127

Thus, the median value of Fibonacci using loop is **0.109 s**

c) **Time taken using recursion and memoisation for 5 observations are as follows:**

```
The time taken by the program on CPU using recursion memoisation is: 0.121
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using recursion memoisation is: 0.098
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using recursion memoisation is: 0.101
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using recursion memoisation is: 0.101
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using recursion memoisation is: 0.089
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

The values in the sorted order are as follows:

0.089s ,0.098s ,0.101s, 0.101s ,0.121s

Thus, the median value of Fibonacci using loop is **0.101s**

d) **Time taken using loop and memoisation for 5 observations are as follows:**

```
The time taken by the program on CPU using loop memoisation is: 0.078
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop memoisation is: 0.108
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop memoisation is: 0.089
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop memoisation is: 0.105
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

```
The time taken by the program on CPU using loop memoisation is: 0.097
PS C:\Users\Sanchit Kanda\Desktop\IIT GN WORK\Semester-2\ES215\Assignments\1\1>
```

The values in the sorted order are as follows:

0.078s ,0.089s ,0.097s, 0.105s ,0.108s

Thus, the median value of Fibonacci using loop is **0.097s**

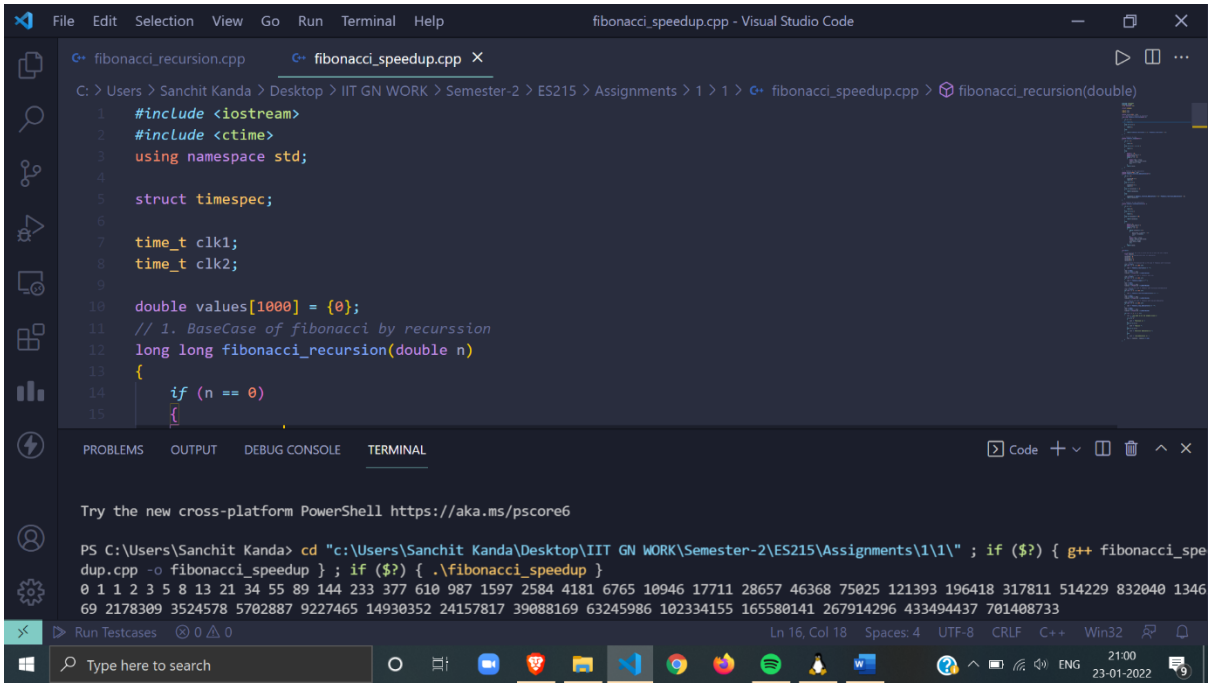
Speedup of all the programs:

a. Speedup for recursion is: **(Time taken using recursion/Time taken using recursion) = 1**

b. Speedup for loop is**: (Time taken using recursion/ Time taken using loop) = 133,605/0.109 = 1225733.95**

c. Speedup for recursion and memoisation is: **133,605/0.101 = (Time taken using recursion/ Time taken using recursion memoisation) = 1322821.78**

d. Speedup for loop memoisation is: **133,605/0.097 = (Time taken using recursion/ Time taken using loop memoisation) = 1377371.13**



I wrote a separate program to calculate the speed up for the individual program but since the Recursion program does not come to completion thus speed up could not be calculated using the code.

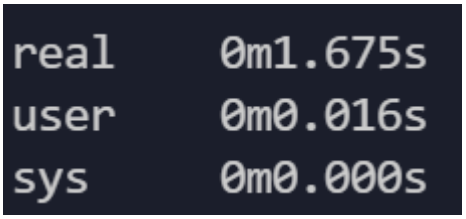
Q2.

A) Output Time

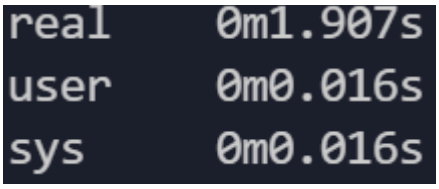
Execution Time = real (in the images showing the data)

a) For C++ using Double

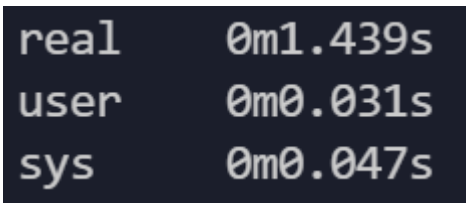
1. **N=32**
   CPU Time=user +sys
   CPU Time = 0.016s + 0.000s = **0.016s**
   System Time =sys= **0.000s**



2. **N=64**
   CPU Time=user +sys
   CPU Time = 0.016s + 0.016s = **0.032s**
   System Time= sys= **0.016s**



3. **N=128**
   CPU Time=user +sys
   CPU Time = 0.031s + 0.047s = **0.078s**
   System Time= sys= **0.047s**



4. **N=256**
   CPU Time=user +sys
   CPU Time = 0.078s + 0.047s = **0.125s**
   System Time= sys= **0.047s**

```
real    0m1.760s
user    0m0.078s
sys     0m0.047s
```

5. **N=512**
   CPU Time=user +sys
   CPU Time = 0.781s + 0.016s = **0.797s**
   System Time= sys= **0.016s**

```
real    0m3.858s
user    0m0.781s
sys     0m0.016s
```

b) For C++ using Integer

1. **N=32**
   CPU Time=user +sys
   CPU Time = 0.000s + 0.016s = **0.016s**
   System Time= sys= **0.016s**

```
real    0m1.438s
user    0m0.000s
sys     0m0.016s
```

2. **N=64**
   CPU Time=user +sys
   CPU Time = 0.016s + 0.000s = **0.016s**
   System Time= sys= **0.000s**

```
real    0m1.026s
user    0m0.016s
sys     0m0.000s
```

3. **N=128**

CPU Time=user +sys
CPU Time = 0.016s + 0.000s = **0.016s**
System Time= sys= **0.000s**

```
real    0m1.267s
user    0m0.016s
sys     0m0.000s
```

4. N=256
   CPU Time=user +sys
   CPU Time = 0.156s + 0.000s = **0.156s**
   System Time= sys= **0.000s**

```
real    0m2.311s
user    0m0.156s
sys     0m0.000s
```

5. N=512
   CPU Time=user +sys
   CPU Time = 0.766s + 0.047s = **0.813s**
   System Time= sys= **0.047s**

```
real    0m3.572s
user    0m0.766s
sys     0m0.047s
```

a) For Python using integer

1. **N=32**

   CPU Time=user +sys

   CPU Time = 0.031s + 0.031s = **0.062s**

   System Time= sys= **0.031s**

```
real    0m3.512s
user    0m0.031s
sys     0m0.031s
```

2. **N=64**

   CPU Time=user +sys

   CPU Time = 0.078s + 0.031s = **0.109s**

   System Time= sys= **0.031s**

```
real    0m3.905s
user    0m0.078s
sys     0m0.031s
```

3. **N=128**

   CPU Time=user +sys

   CPU Time = 0.484s + 0.000s = **0.484s**

   System Time= sys= **0.000s**

```
real    0m2.539s
user    0m0.484s
sys     0m0.000s
```

4. **N=256**

   CPU Time=user +sys

   CPU Time = 3.406s + 0.047s = **3.453s**

   System Time= sys= **0.047s**

```
real    0m5.792s
user    0m3.406s
sys     0m0.047s
```

5. **N=512**
   CPU Time=user +sys
   CPU Time = 30.344s + 0.078s = **30.422s**
   System Time= sys= **0.078s**

```
real    0m32.467s
user    0m30.344s
sys     0m0.078s
```

b) For Python using Double

1. **N=32**
   CPU Time=user +sys
   CPU Time = 0.047s + 0.016s = **0.063s**
   System Time= sys= **0.016s**

```
real    0m1.257s
user    0m0.047s
sys     0m0.016s
```

2. **N=64**
   CPU Time=user +sys
   CPU Time = 0.078s + 0.031s = **0.141s**
   System Time= sys= **0.031s**

```
real    0m1.555s
user    0m0.078s
sys     0m0.031s
```

3. **N=128**
   CPU Time=user +sys
   CPU Time = 0.406s + 0.078s = **0.484s**
   System Time= sys= **0.078s**

```
real    0m2.119s
user    0m0.406s
sys     0m0.078s
```

4. **N=256**
   CPU Time=user +sys
   CPU Time = 3.406s + 0.031s = **3.437s**
   System Time= sys= **0.031s**

```
real    0m4.258s
user    0m3.406s
sys     0m0.031s
```

5. **N=512**
   CPU Time=user +sys
   CPU Time = 30.641s + 0.125s = **30.766s**
   System Time= sys= **0.125s**

```
real    0m32.619s
user    0m30.641s
sys     0m0.125s
```

B) Using Language Hooks

a) C++ program for integer

1. N=32

Execution Time: **0.931s**

Meat Portion: **0.03125s**

(Meat Portion/Execution Time): 0.0335

```
The time taken by the meat portion of the program is: 0.03125

real    0m0.931s
user    0m0.000s
sys     0m0.047s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT GN WORK/Semester-2/ES215/Assignments/1/2$
```

2. N=64

Execution Time: **1.641s**

Meat Portion: **0.015625s**

(Meat Portion/Execution Time): 0.009521

```
The time taken by the meat portion of the program is: 0.015625

real    0m1.641s
user    0m0.016s
sys     0m0.000s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT GN WORK/Semester-2/ES215/Assignments/1/2$ time ./a.out
```

3. **N=128**

Execution Time: **1.745s**

Meat Portion: **0.015625s**

(Meat Portion/Execution Time): 0.0089

```
The time taken by the meat portion of the program is: 0.015625

real    0m1.745s
user    0m0.016s
sys     0m0.031s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT GN WORK/Semester-2/ES215/Assignments/1/2$
```

4. **N=256**

Execution Time: **2.150s**

Meat Portion: **0.09375s**

(Meat Portion/Execution Time): 0.043

```
The time taken by the meat portion of the program is: 0.09375

real    0m2.150s
user    0m0.063s
sys     0m0.031s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT
```

5. N=512

Execution Time: **6.323s**

Meat Portion: **0.84375s**

(Meat Portion/Execution Time): 0.133

```
The time taken by the meat portion of the program is: 0.84375

real    0m6.323s
user    0m0.797s
sys     0m0.063s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT
```

b) C++ program for double

1. N=32

Execution Time: **1.278s**

Meat Portion**: 0.015625s**

(Meat Portion/Execution Time): 0.1222

```
The time taken by the meat portion of the program is: 0.015625

real    0m1.278s
user    0m0.000s
sys     0m0.016s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT G
```

2. N=64

Execution Time: **0.932s**

Meat Portion: **0.015625s**

(Meat Portion/Execution Time): 0.0167

```
The time taken by the meat portion of the program is: 0.015625

real    0m0.932s
user    0m0.000s
sys     0m0.016s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT G
```

3. N=128

Execution Time: **1.373s**

Meat Portion: **0.015625s**

(Meat Portion/Execution Time): 0.0113

```
The time taken by the meat portion of the program is: 0.015625

real    0m1.373s
user    0m0.000s
sys     0m0.016s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT G
```

4. N=256

Execution Time: **3.026s**

Meat Portion: **0.171875s**

(Meat Portion/Execution Time): 0.056799

```
The time taken by the meat portion of the program is: 0.171875

real    0m3.026s
user    0m0.141s
sys     0m0.031s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT G
```

5. N=512

Execution Time: **3.894s**

Meat Portion: **1.0156s**

(Meat Portion/Execution Time): 0.2608

```
The time taken by the meat portion of the program is: 1.01562

real    0m3.894s
user    0m0.969s
sys     0m0.078s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Kanda/Desktop/IIT
```

a) Python program for Double

1. N=32

Execution Time: **1.126s**

Meat Portion: **0.0110s**

(Meat Portion/Execution Time): 0.00976

```
Meat portion time :  0.011014700000048721

real    0m1.126s
user    0m0.047s
sys     0m0.000s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchit Ka
```

2. N=64

Execution Time: **0.806s**

Meat Portion: **0.05487s**

(Meat Portion/Execution Time): 0.067

```
Meat portion time :  0.05487369999916425

real    0m0.806s
user    0m0.094s
sys     0m0.000s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanchi
```

3. N=128

Execution Time: **1.769s**

Meat Portion: **0.403s**

(Meat Portion/Execution Time): 0.2278

```
Meat portion time :  0.4038940000000366

real    0m1.769s
user    0m0.438s
sys     0m0.000s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanch
```

4. N=256

Execution Time: **4.882s**

Meat Portion: **3.392s**

(Meat Portion/Execution Time): 0.69

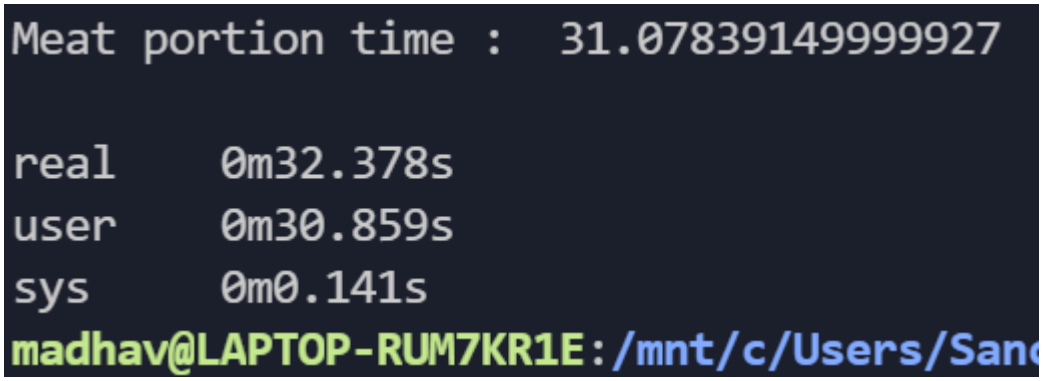
```
Meat portion time :  3.3926918999995905

real    0m4.882s
user    0m3.438s
sys     0m0.031s
madhav@LAPTOP-RUM7KR1E:/mnt/c/Users/Sanch
```

5. N=512

Execution Time: **32.378s**

Meat Portion: **31.07s**

(Meat Portion/Execution Time): 0.95



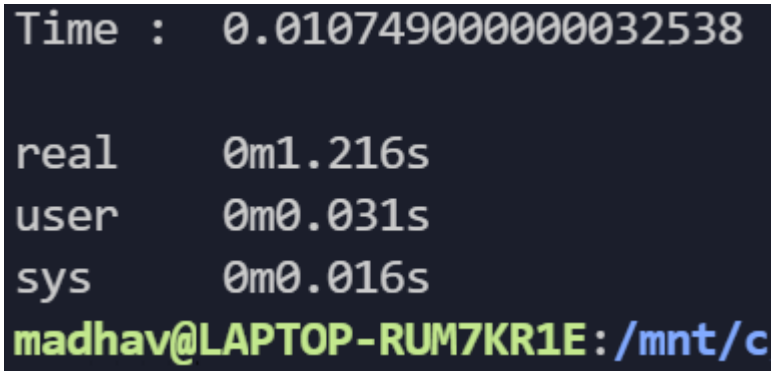
b) Python program for Integer

1. **N=32**

Execution Time: **1.216s**

Meat Portion: **0.0107s**

(Meat Portion/Execution Time): 0.087



2. N=64

Execution Time: **1.736s**

Meat Portion: **0.0566s**

(Meat Portion/Execution Time): 0.032

```
Time :   0.0566709999936909

real     0m1.736s
user     0m0.063s
sys      0m0.031s
```

3. N=128

Execution Time: **2.135s**

Meat Portion: **0.599s**

(Meat Portion/Execution Time): 0.28

```
Time :   0.5997846000000209

real     0m2.135s
user     0m0.688s
sys      0m0.047s
madhav@LAPTOP-RUM7KR1E:/mnt/
```

4. N=256

Execution Time**: 5.590s**

Meat Portion: **4.251s**

(Meat Portion/Execution Time): **0.76**

```
Time :   4.251851899999565

real     0m5.590s
user     0m4.281s
sys      0m0.078s
madhav@LAPTOP-RUM7KR1E:/mnt/c/
```

5. N=512

Execution Time: **39.384s**

Meat Portion: **38.05s**

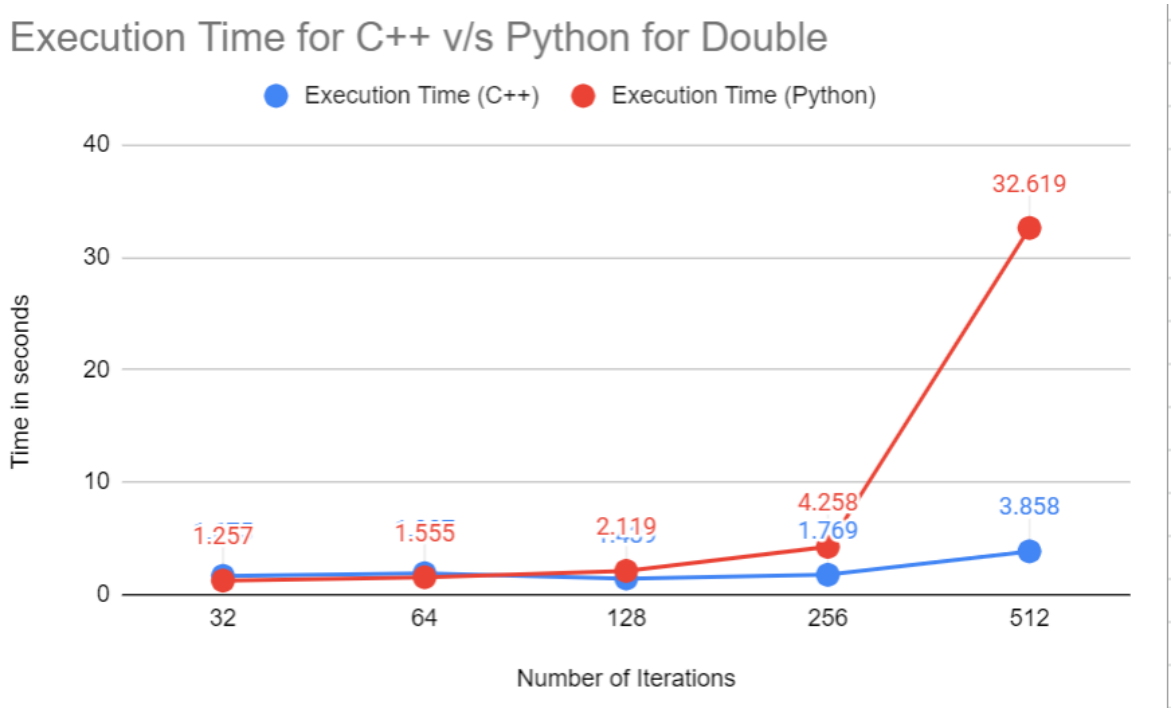
(Meat Portion/Execution Time): 0.96

```
Time :   38.051534600000195

real    0m39.384s
user    0m37.813s
sys     0m0.125s
madhav@LAPTOP-RUM7KR1E:/mnt/c/
```

c) Plots for Execution Time

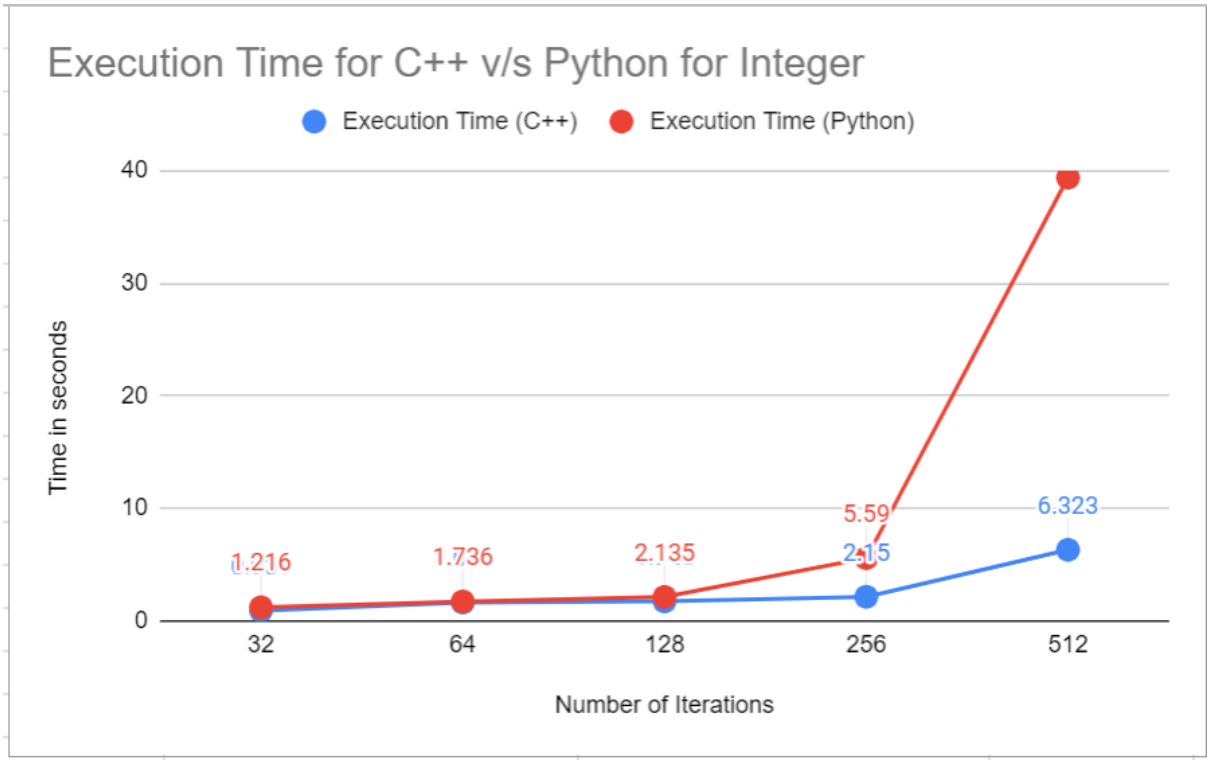
1. Execution Time for Double of the Two programming languages

Number of Iterations	Execution Time (C++)	Execution Time (Python)
32	1.675	1.257
64	1.907	1.555
128	1.439	2.119
256	1.769	4.258
512	3.858	32.619



2. Execution Time for Integer of the Two programming languages

Number of Iterations	Execution Time (C++)	Execution Time (Python)
32	0.931	1.216
64	1.641	1.736
128	1.745	2.135
256	2.15	5.59
512	6.323	39.384



Observations

According to the above graphs, it is clear that the execution time for python in both Integer and Double matrix multiplication is higher than that of C++.

This also goes well with the fact that Python is a high-level language whereas C++ is middle-level language and execution time for high level language is more than the execution time on a middle-level language given the same program.

From the data mentioned in the a) part it is clear that the system time (real) is less than the program execution time for any program on any language. This is because execution time takes into consideration the time taken from the start to finish of the call. Thus it also includes the time that we take to give the input. System time is the time spent in the kernel within the process. Since execution time includes the system time as it calculates the time from start to finish of the program thus, it is always greater than system time.