Name: Siddhesh Kanawade
Roll number: 20110199

ES215: COMPUTER ORGANISATION AND ARCHITECTURE

(Assignment -1)

[Github Repository]

Question 1

Here, I have found the program execution time for two cases:

1. To print fibonacci series till the 55th term.
2. To print fibonacci series till the 100th term.

Method used	Till 55th term(in ns)	TIll 100th term(in ns)	Till 55th term(in sec)	TIll 100th term(in sec)
Using recursion	1266861511155	1.744402979 * 10^22	1266.861511	1.744402979 * 10^13
Using loop	58208	108009	0.000058208	0.000108009
Using recursion and memoization	53669	110637	0.000053669	0.000110637
Using loop and memoization	58843	125665	0.000058843	0.000125665

Note: *I was not able to calculate the time for printing till the 100th term using recursion practically since it was taking longer than 12 hours to execute. After calculating the theoretical value of execution time for printing till the 100th term using recursion it was found that it can't be executed with the laptop configuration I have. Thus, I have calculated the theoretical value of execution time to print fibonacci series till the 100th term.*

Calculation of execution time to print till 100th term using recursion.
Time taken to print till 55th term = 1266861511155 nanoseconds.
Time complexity of program = O(1.68^n)

Assumption: TIme required to execute individual steps remains constant with number of iterations over function.

Thus,
Total time required = Time taken to print till 55th term * 1.68^45
= 1266861511155 * 1.68^45
= **1.744402979 * 10^22 nanoseconds**
= **1.744402979 8 10^13 seconds.**

Speedup of all programs(keeping program(1) as reference**):**

Method used	Speedup(till 55th term)	Speedup(till 100th term)
Using recursion	1	1
Using loop	0.0000000459466165	0
Using recursion and memoization	0.00000004236374658	0
Using loop and memoization	0.00000004644785518	0

From the first table, we can see that time taken for execution of the 100th term using recursion is infinitesimal in comparison to time taken by other methods. Thus, when calculated in a sheet, the ratio is very small and hence the system returns zero.

Question 2

Programming languages selected:
Bucket1: **C++**
Bucket2: **Python**

(a) When datatype = **Integer**

a) The system time and CPU time can be calculated by executing the following commands:

C++: *g++ filename.cpp -o outputFilename*
time ./outputfilename

Python: *time python3 outputfilename.py*

By executing the time command we get three time values: user, real and system.
Real: It's the time spent including all time slices used by other processes and time processes it spends blocked(for example: time elapsed in taking input). Thus, this time is also claimed to be the total **program execution time.**
User: It's the time spent *outside* the kernel for a given program execution. It doesn't consider the time spent in other processes and time spent when the system was blocked.
Sys: It's the amount of time spent *within* the kernel process. Similar to user time, it too doesn't consider the other time elapses.

CPU time = User + Sys

Following is the execution time of the program with various iterations

Language	N	User(in sec)	Sys(in sec)	Program execution(in sec)	CPU(in sec)
Python	32	0.04	0.004	0.046	0.044
Python	64	0.057	0.004	0.061	0.061
Python	128	0.307	0.004	0.323	0.311
Python	256	2.038	0.012	2.086	2.05
Python	512	15.945	0.028	16.056	15.973
C++	32	0.005	0	0.005	0.005
C++	64	0.005	0	0.006	0.005
C++	128	0.017	0	0.035	0.017
C++	256	0.081	0.008	0.124	0.089
C++	512	0.665	0.016	0.78	0.681

It can be noted that the execution time of the program scripted in Python is larger than the program scripted in C++. Thus, we can claim that C++ has less execution time than Python and hence, *C++ is faster than Python*.

b) **Language hook:** It's the time elapsed between the part of code which is responsible for multiplication of matrices.

Language	N	Program execution(in sec)
Python	32	0.046
Python	64	0.061
Python	128	0.323
Python	256	2.086
Python	512	16.056
C++	32	0.005
C++	64	0.006
C++	128	0.035
C++	256	0.124
C++	512	0.78

Partwise execution time(in nano sec)	conversion of nano sec to sec	Partwise execution time(in sec)	Ratio
-		0.005210243999	0.1132661739
-		0.035584325	0.5833495902
-		0.258985917	0.8018139845
-		1.963270933	0.9411653562
-		15.71875391	0.9789956347
412010	1000000000	0.00041201	0.082402
2102190	1000000000	0.00210219	0.350365
9967053	1000000000	0.009967053	0.2847729429
76712473	1000000000	0.076712473	0.6186489758
647789033	1000000000	0.647789033	0.8304987603

Ratio = *Part Wise execution time / Program execution time*

Observations:

The execution time for python when N = 512 is about 15.7 sec, which is much larger than other observations. Thus, chosening python for higher values of N will not be an optimum choice.

The ratio gets closer to one as the value of N increases, this is because the contribution of sys time remains almost constant for all iterations.

Due to the high value of program execution time and part-wise execution time, the ratio of python at N = 512 is much closer to one.

(b) When datatype = **Double**

Assumption: *Python doesn't have any datatype named "double", hence float has been used as double datatype in python.*

a) The terms and definitions remain the same as in part (a).
CPU time = User + Sys

Following is the execution time of the program with various iterations

Language	N	User(in sec)	Sys(in sec)	Program execution(in sec)	CPU(in sec)
Python	32	0.018	0.014	0.041	0.032
Python	64	0.073	0.021	0.145	0.094
Python	128	0.337	0.013	0.394	0.35
Python	256	2.173	0.02	2.39	2.193
Python	512	17.826	0.068	18.707	17.894
C++	32	0.007	0	0.007	0.007
C++	64	0.006	0	0.015	0.006
C++	128	0.023	0.004	0.039	0.027
C++	256	0.141	0.004	0.169	0.145
C++	512	0.875	0.016	1.004	0.891

Observations:

1. C++ is faster than Python.
2. The execution time has increased in both C++ and Python compared to the execution time when datatype was *integer.* The major difference can be seen in iteration where N = 512.

b) Language hook: It's the time elapsed between the part of code which is responsible for multiplication of matrices.

Language	N	Program execution(in sec)
Python	32	0.041
Python	64	0.145
Python	128	0.394
Python	256	2.39
Python	512	18.707
C++	32	0.007
C++	64	0.015
C++	128	0.039
C++	256	0.169
C++	512	1.004

Partwise execution time(in nano sec)	conversion of nano sec to sec	Partwise execution time(in sec)	Ratio
-		0.005592743	0.1364083659
-		0.036287263	0.2502569862
-		0.263145456	0.667881868
-		2.013467079	0.8424548448
-		17.30866712	0.9252508218
415329	1000000000	0.000415329	0.05933271429
1355981	1000000000	0.001355981	0.09039873333
10989354	1000000000	0.010989354	0.2817783077
96217906	1000000000	0.096217906	0.5693367219
721825262	1000000000	0.721825262	0.7189494641

Ratio = *Part Wise execution time / Program execution time*

Observations:

1. The execution time for python when N = 512 is about 18.7 sec, which is much larger than other observations. Thus, chosening python for higher values of N will not be an optimum choice.

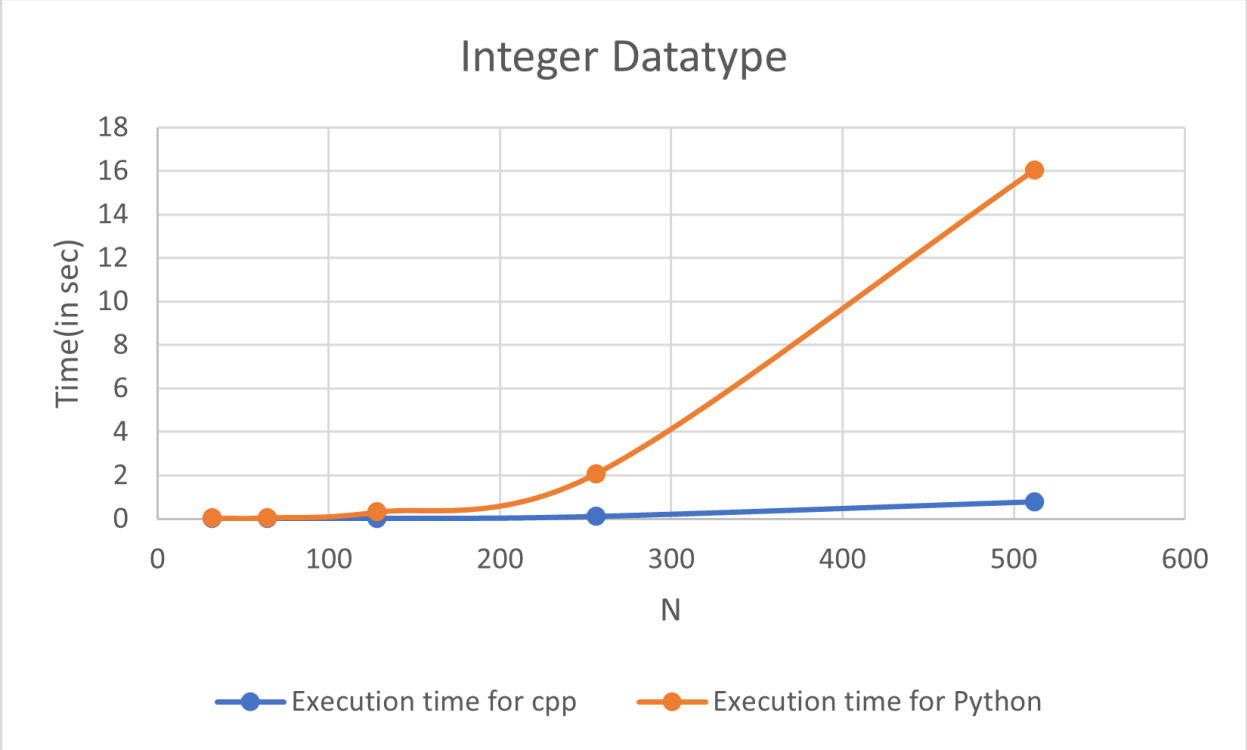
2. The ratio gets closer to one as the value of N increases, this is because the contribution of sys time remains almost constant for all iterations.
3. Due to the high value of program execution time and part-wise execution time, the ratio of python at N = 512 is much closer to one.
4. The execution time increases due to change in datatype from integer to double.

---

c) Plots:

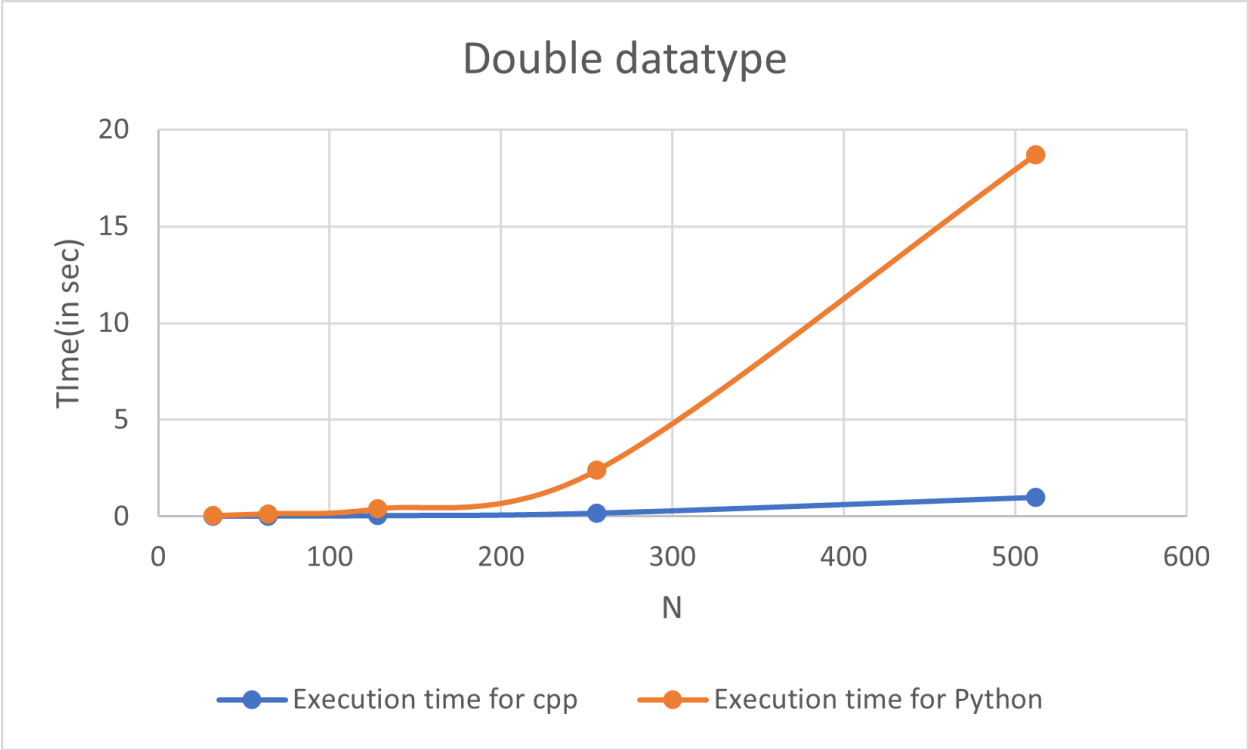
***Program Execution time vs N**:*

1. **For Integer:**



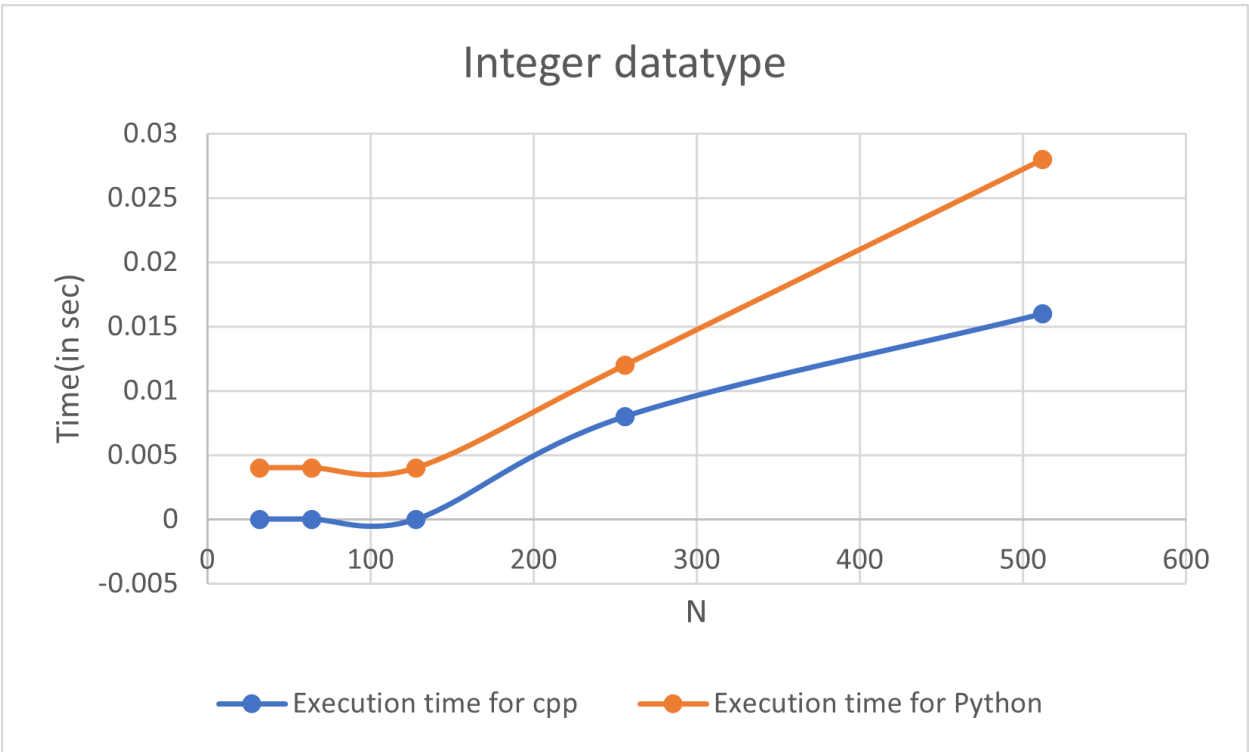
:

2. **For double:**

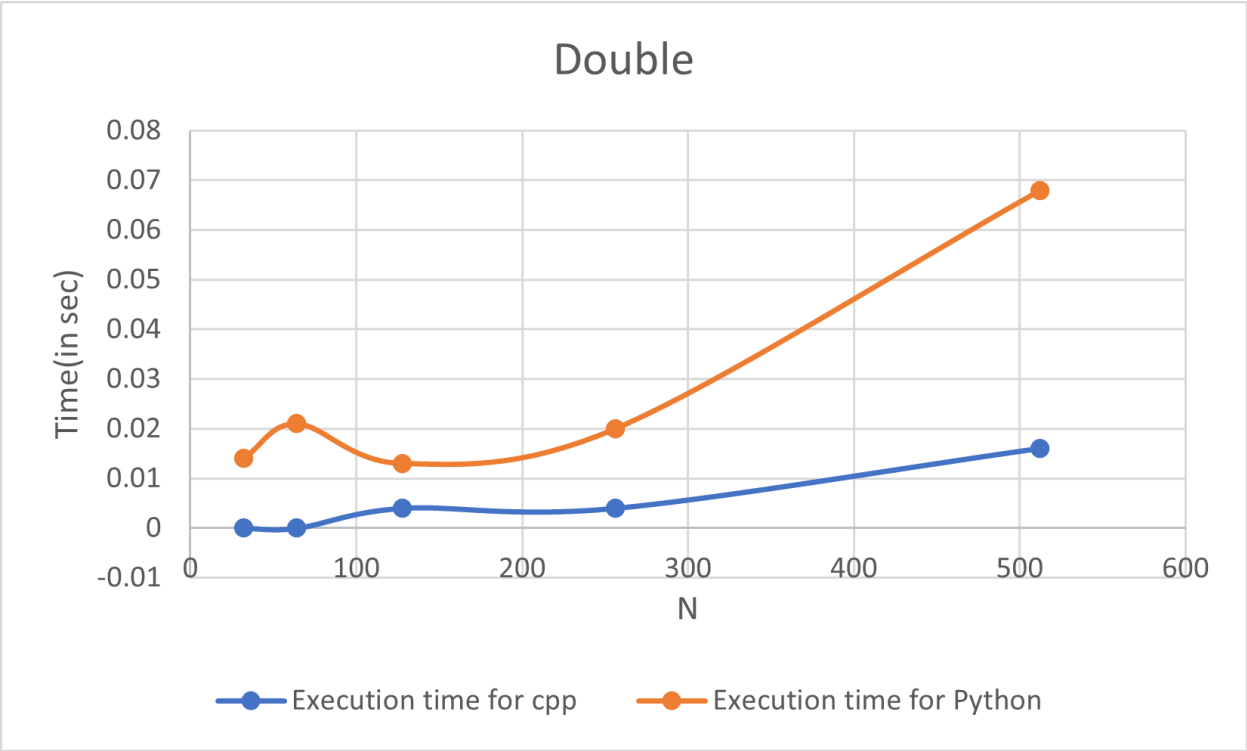


***System execution time vs N**:*

1. **Integer**:



2. **Double:**

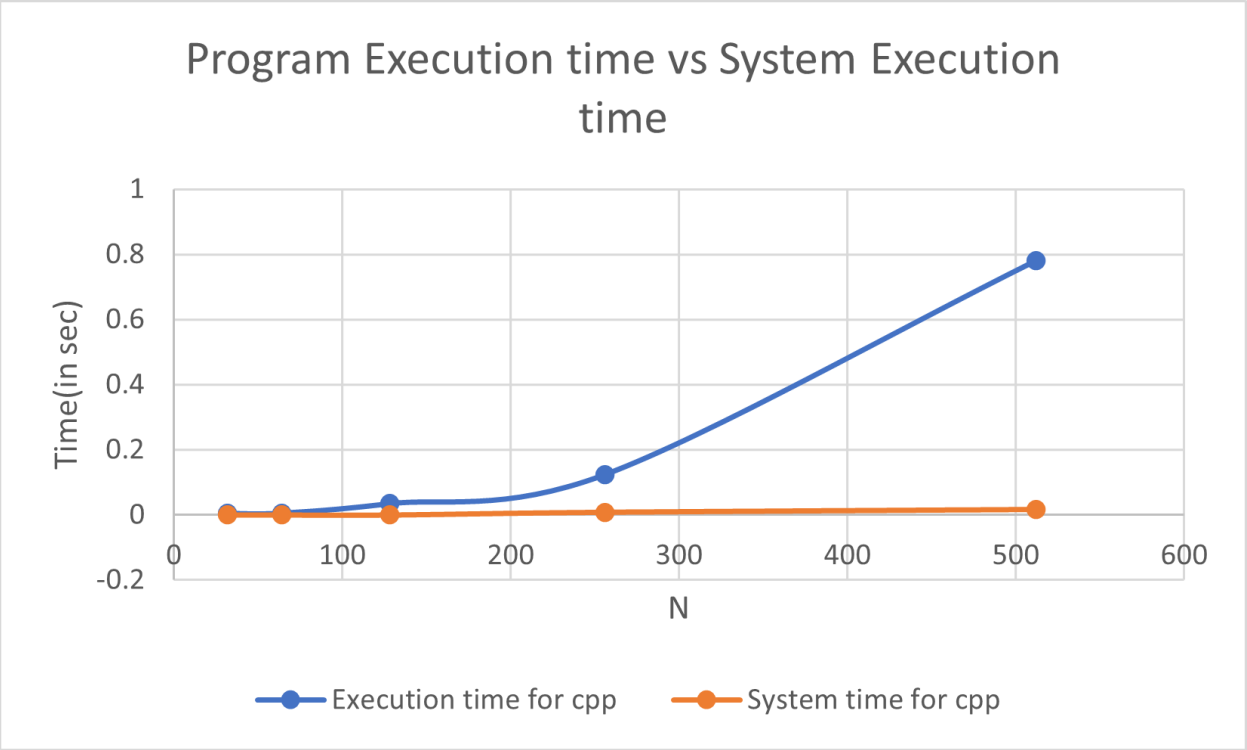


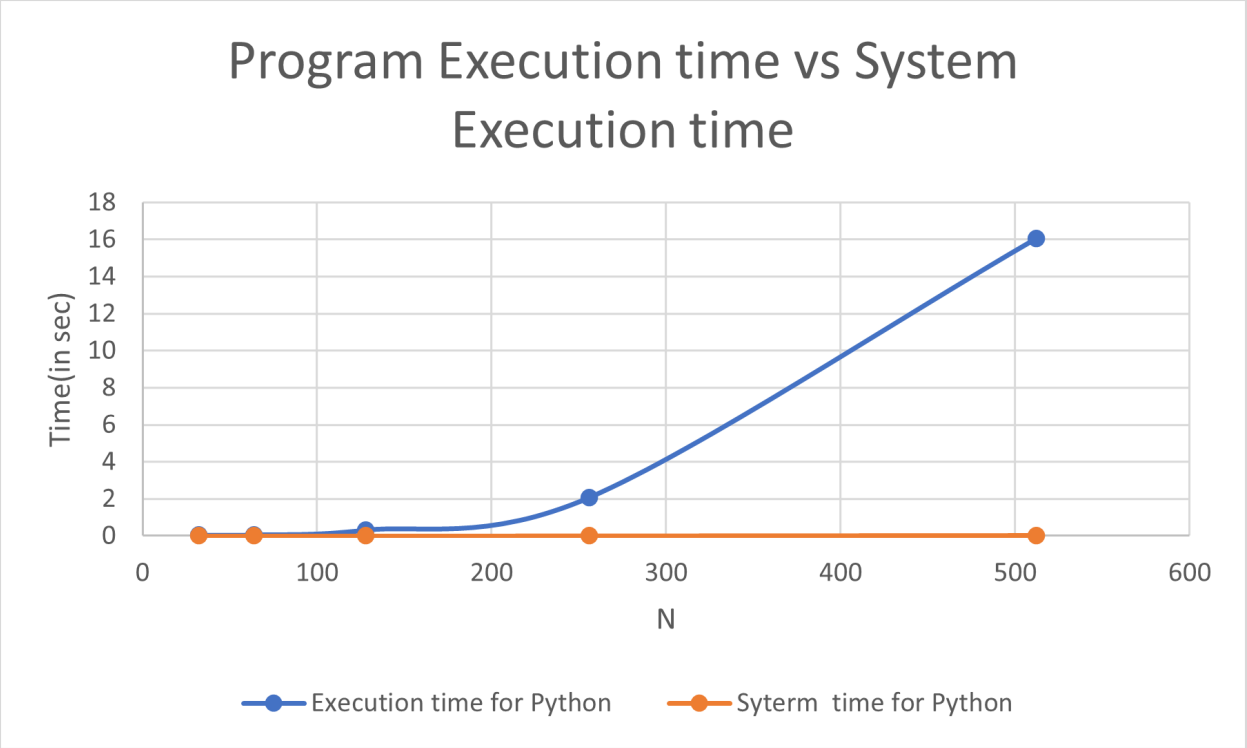
Observations(combined):

1. Program execution time for both C++ and Python increases as N increases. Program execution time for Python increases drastically as N moves from 256 to 512, whereas, the increase in program execution time for C++ is gradual.
2. Program execution time for double is more than that for integer.
3. Program execution time is comparable for both Python and C++ for lower values of N.
4. System execution time for Python is more than that for C++
5. Generally, system execution time increases with increase in N.

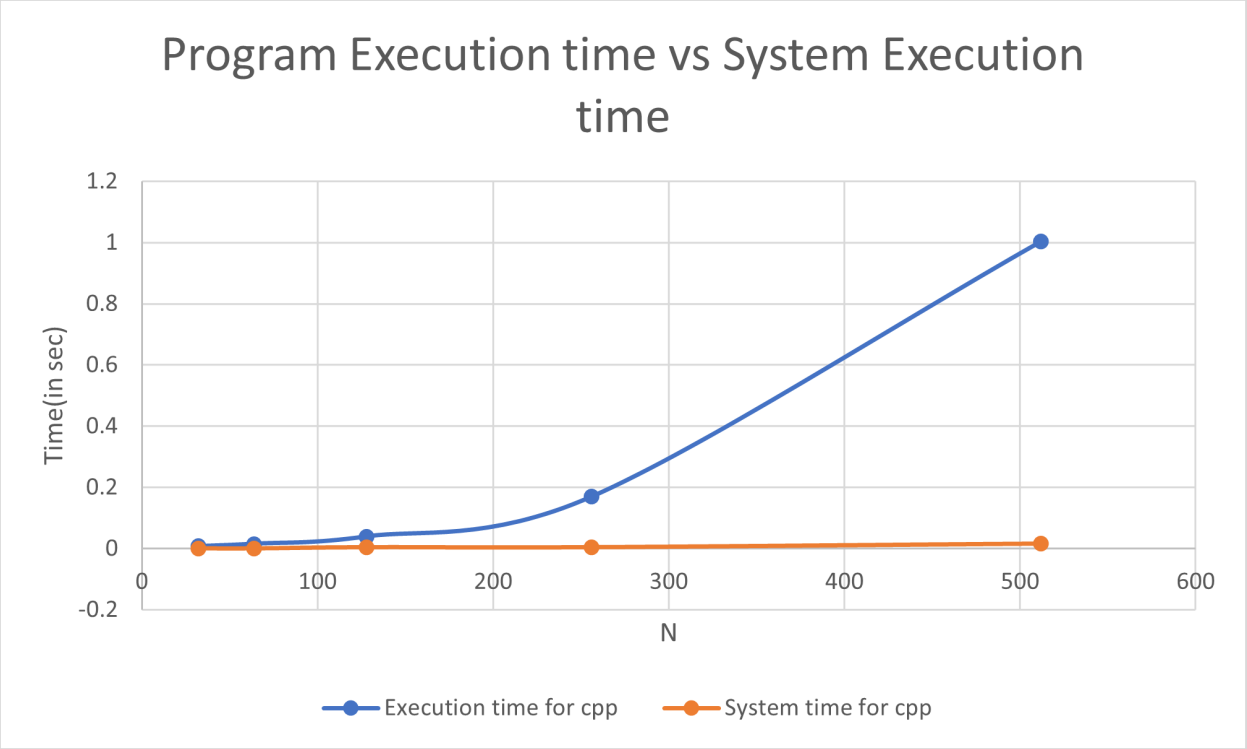
Comparison for Program Execution time VS System Execution time:

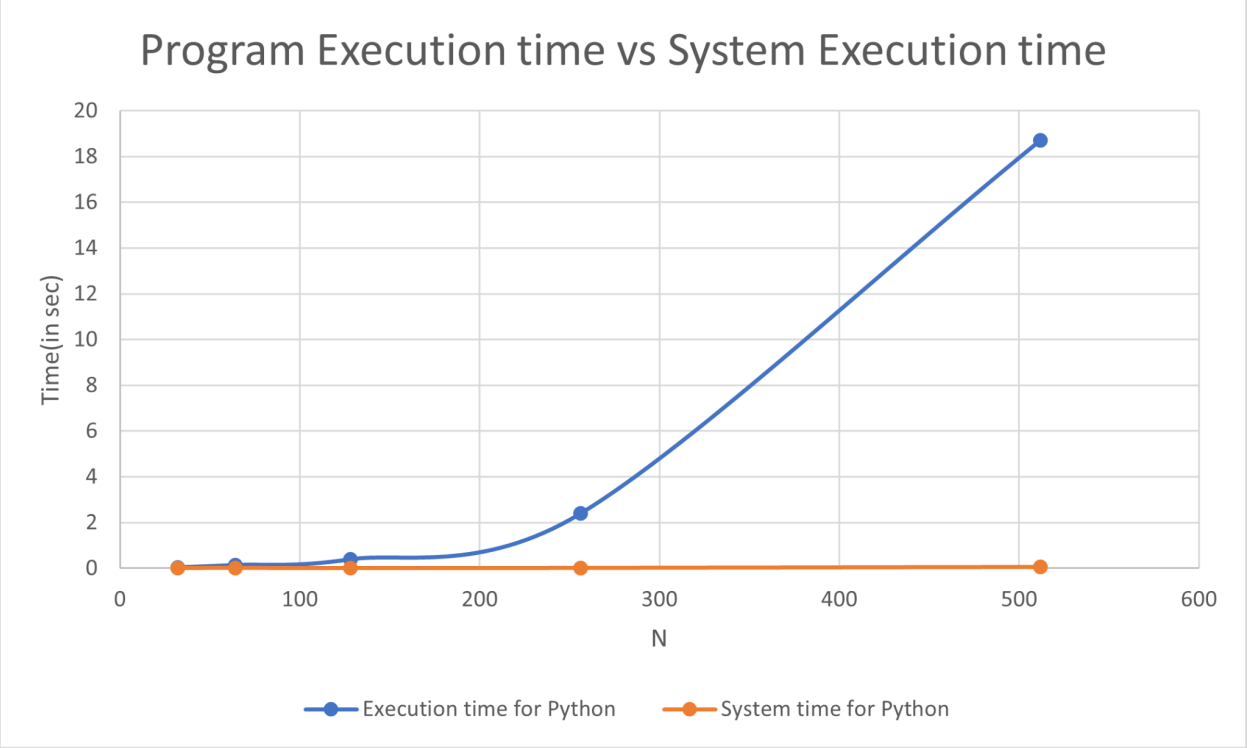
1. **Integer:**





2. Double:


Program Execution time vs System Execution time
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Time(in sec)
0
100
200
300
400
500
600
N
Execution time for cpp
System time for cpp


Program Execution time vs System Execution time
20
18
16
14
12
10
8
6
4
2
0
Time(in sec)
0
100
200
300
400
500
600
N
Execution time for Python
System time for Python

Observations:
System execution time is much lesser than Program execution time and it becomes negligible for higher values of N.