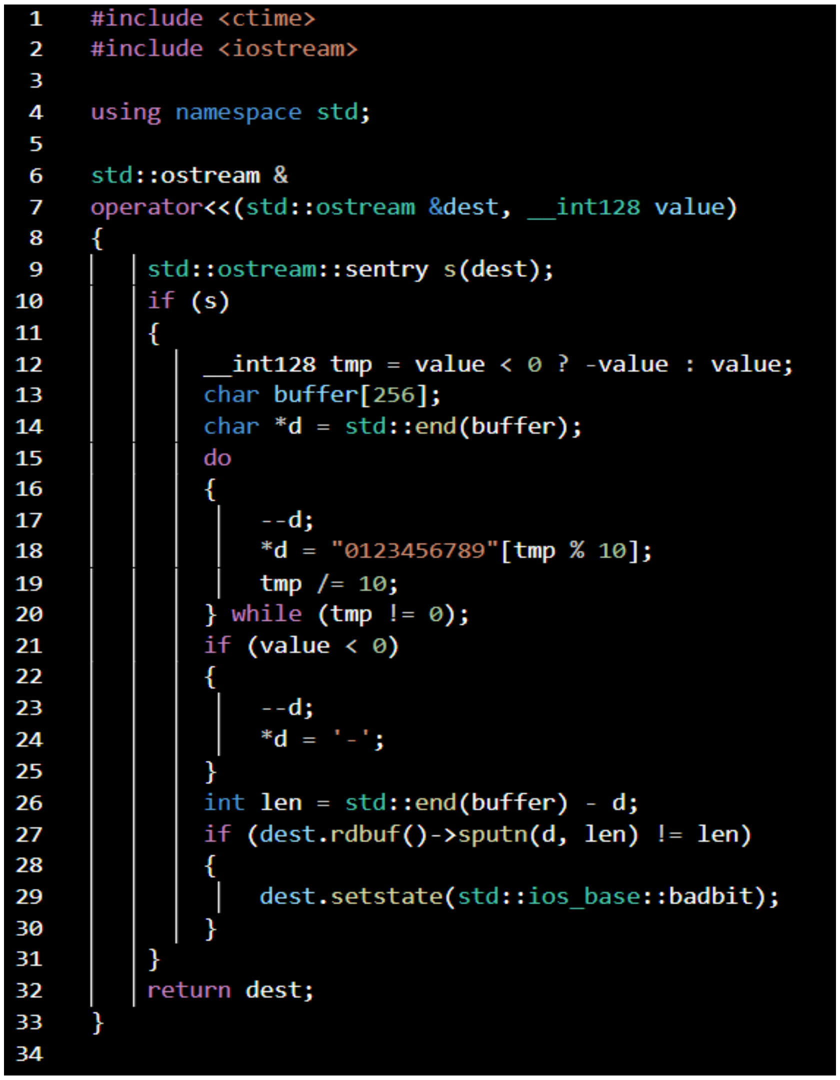
ES215: Semester II [2021-2022]

Assignment-1

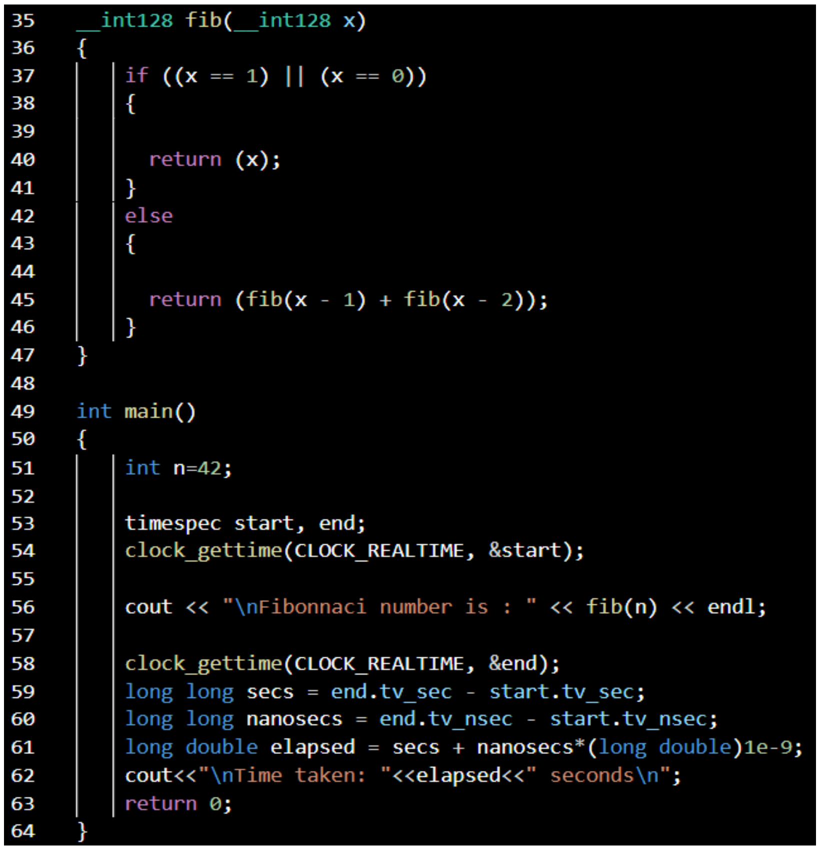
1) Here, I have assumed 0 to be the first Fibonacci number i.e., F(0) be the 1st Fibonacci number. Therefore, F(n) is denoted as the (n+1)th Fibonacci number.

a) Initially, I tried to execute for n=99 (which is for the first 100 Fibonacci numbers). But since the order of the recursive program is O((1.6180)^n), after a specific value of n, the value of the time taken of the program will increase rapidly. Therefore, I have taken n=42 and then derive the time taken for the execution of n=99.

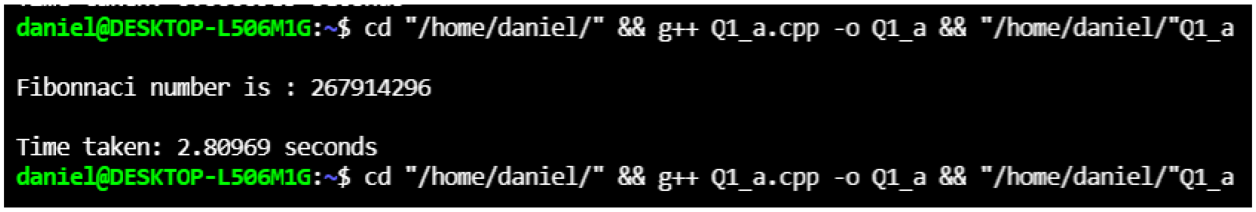
Code:



```
#include <ctime>
#include <iostream>

using namespace std;

std::ostream &
operator<<(std::ostream &dest, __int128 value)
{
    std::ostream::sentry s(dest);
    if (s)
    {
        __int128 tmp = value < 0 ? -value : value;
        char buffer[256];
        char *d = std::end(buffer);
        do
        {
            --d;
            *d = "0123456789"[tmp % 10];
            tmp /= 10;
        } while (tmp != 0);
        if (value < 0)
        {
            --d;
            *d = '-';
        }
        int len = std::end(buffer) - d;
        if (dest.rdbuf()->sputn(d, len) != len)
        {
            dest.setstate(std::ios_base::badbit);
        }
    }
    return dest;
}

```



```cpp
__int128 fib(__int128 x)
{
    if ((x == 1) || (x == 0))
    {

      return (x);
    }
    else
    {

      return (fib(x - 1) + fib(x - 2));
    }
}

int main()
{
    int n=42;

    timespec start, end;
    clock_gettime(CLOCK_REALTIME, &start);

    cout << "\nFibonnaci number is : " << fib(n) << endl;

    clock_gettime(CLOCK_REALTIME, &end);
    long long secs = end.tv_sec - start.tv_sec;
    long long nanosecs = end.tv_nsec - start.tv_nsec;
    long double elapsed = secs + nanosecs*(long double)1e-9;
    cout<<"\nTime taken: "<<elapsed<<" seconds\n";
    return 0;
}
```



```
daniel@DESKTOP-L506M1G:~$ cd "/home/daniel/" && g++ Q1_a.cpp -o Q1_a && "/home/daniel/"Q1_a

Fibonnaci number is : 267914296

Time taken: 2.80969 seconds
daniel@DESKTOP-L506M1G:~$ cd "/home/daniel/" && g++ Q1_a.cpp -o Q1_a && "/home/daniel/"Q1_a
```

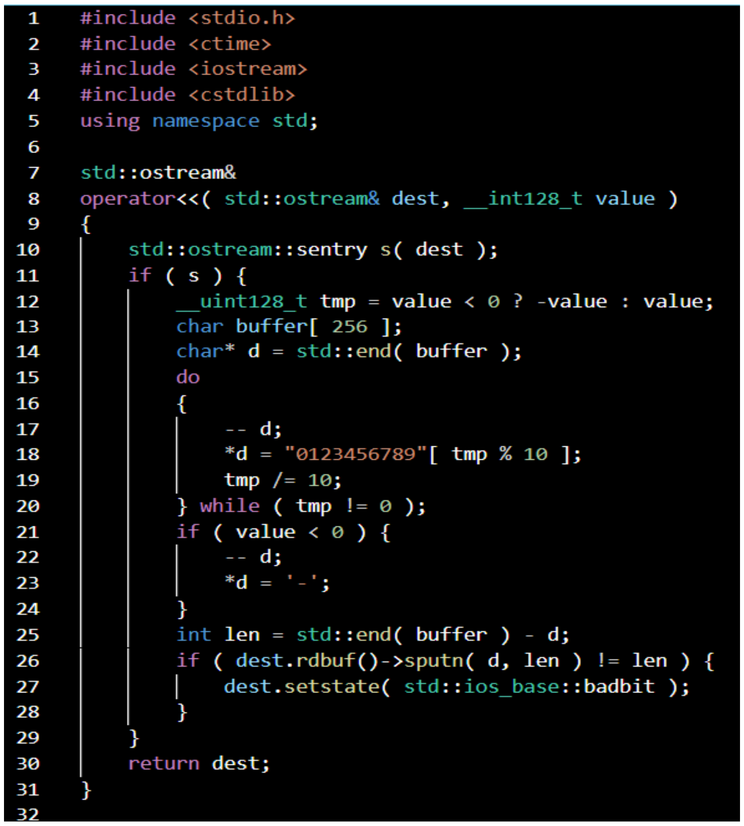
Time taken for n=42 is 2.80969 seconds. Since t(n) = A*(1.6180^n), we substitute n=42 here to get the value of the constant A.

Therefore, A=2.81/(1.6180^42)

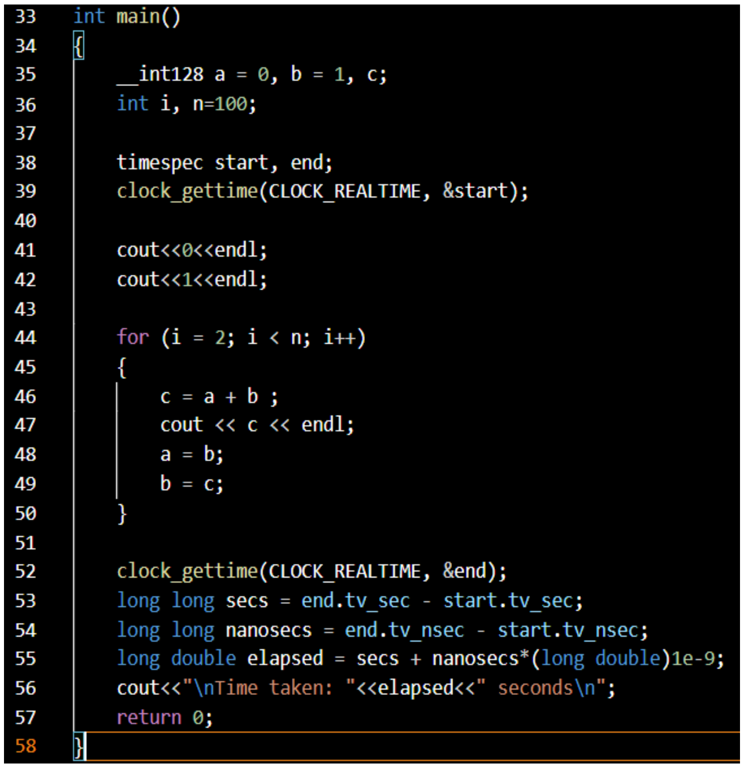
Thus, the time taken for n=99 (which leads to 100 Fibonacci numbers as per our assumption) = **2.81*(1.6180^57) seconds.** (Let's consider this t(a))

Speedup = t(a)/t(a) = **1**

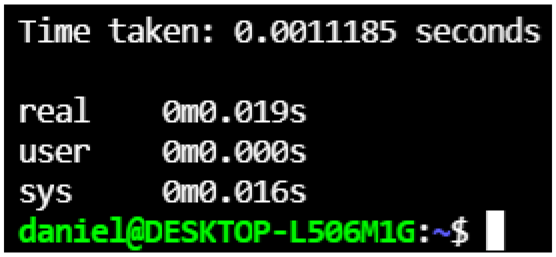
b) Code:



```cpp
#include <stdio.h>
#include <ctime>
#include <iostream>
#include <cstdlib>
using namespace std;

std::ostream&
operator<<( std::ostream& dest, __int128_t value )
{
    std::ostream::sentry s( dest );
    if ( s ) {
        __uint128_t tmp = value < 0 ? -value : value;
        char buffer[ 256 ];
        char* d = std::end( buffer );
        do
        {
            -- d;
            *d = "0123456789"[ tmp % 10 ];
            tmp /= 10;
        } while ( tmp != 0 );
        if ( value < 0 ) {
            -- d;
            *d = '-';
        }
        int len = std::end( buffer ) - d;
        if ( dest.rdbuf()->sputn( d, len ) != len ) {
            dest.setstate( std::ios_base::badbit );
        }
    }
    return dest;
}

```



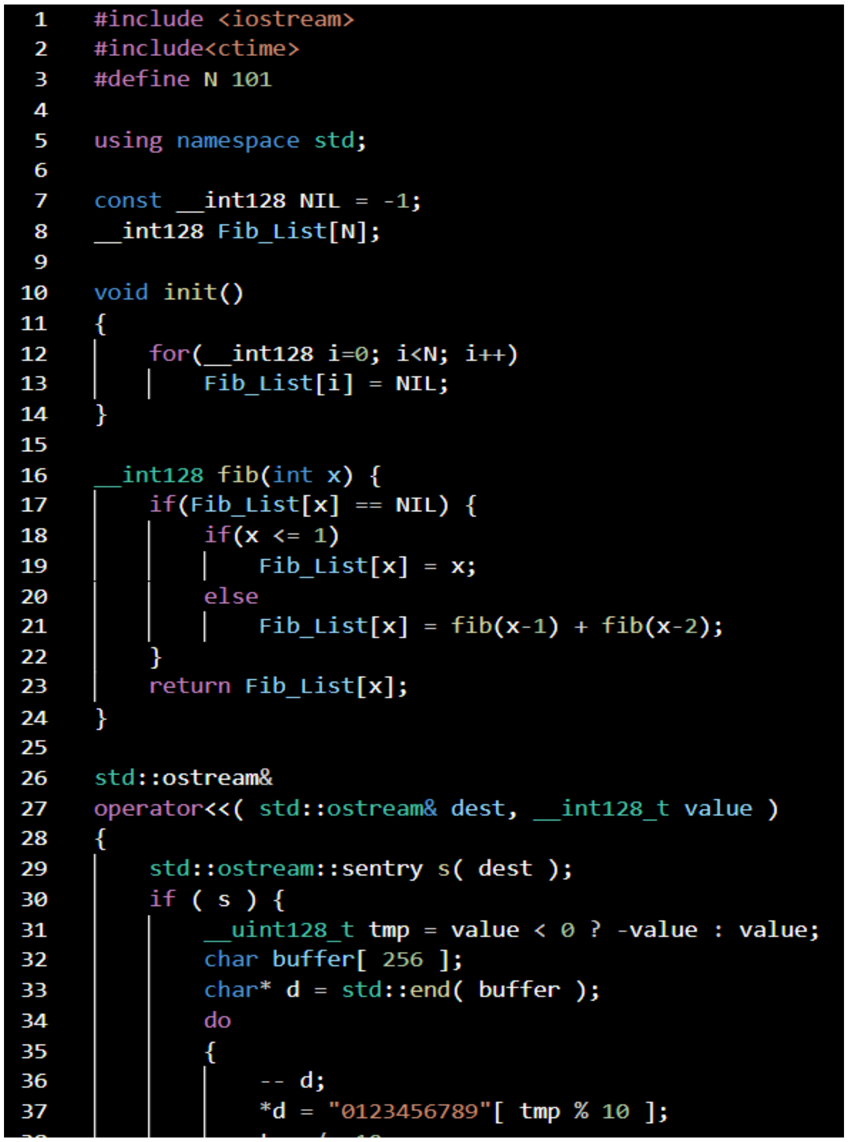
```cpp
int main()
{
    __int128 a = 0, b = 1, c;
    int i, n=100;

    timespec start, end;
    clock_gettime(CLOCK_REALTIME, &start);

    cout<<0<<endl;
    cout<<1<<endl;

    for (i = 2; i < n; i++)
    {
        c = a + b ;
        cout << c << endl;
        a = b;
        b = c;
    }

    clock_gettime(CLOCK_REALTIME, &end);
    long long secs = end.tv_sec - start.tv_sec;
    long long nanosecs = end.tv_nsec - start.tv_nsec;
    long double elapsed = secs + nanosecs*(long double)1e-9;
    cout<<"\nTime taken: "<<elapsed<<" seconds\n";
    return 0;
}
```



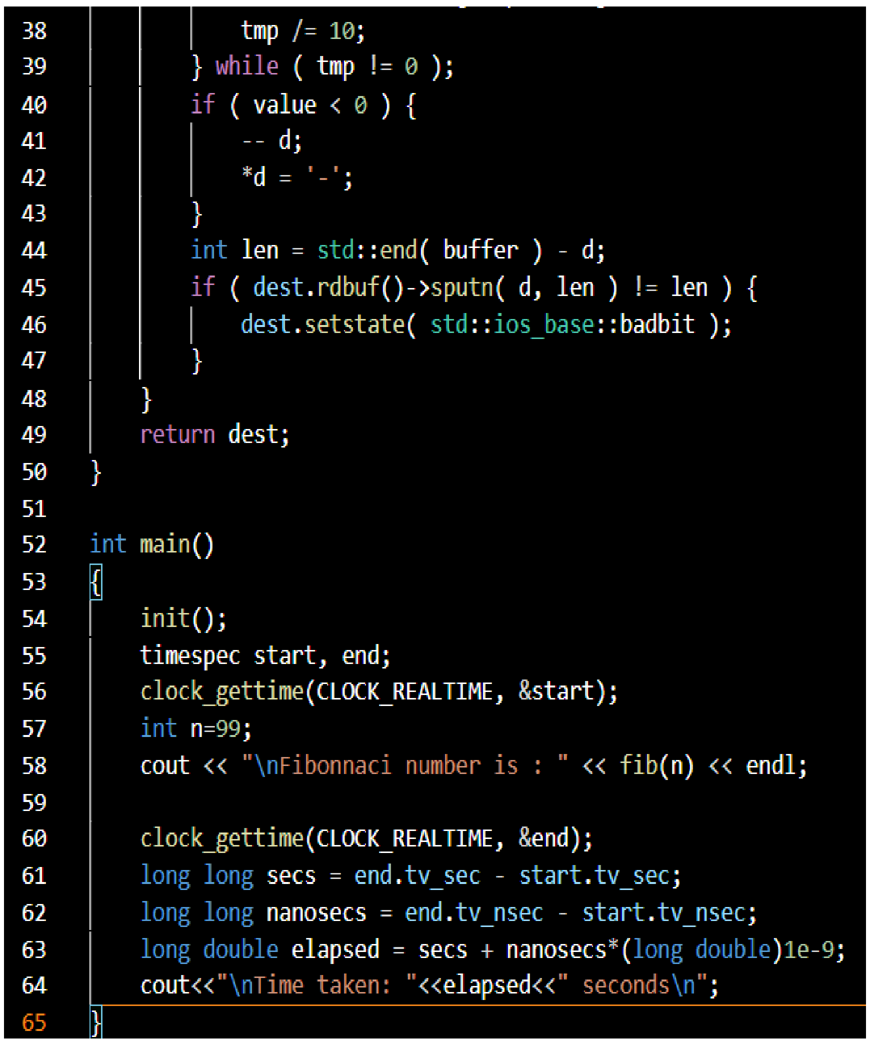
```
Time taken: 0.0011185 seconds

real    0m0.019s
user    0m0.000s
sys     0m0.016s
daniel@DESKTOP-L506M1G:~$
```

Time taken t(b) = **0.0011185 seconds.**
Speedup = t(a)/t(b) = **2512.2932498*(1.6180^57)**

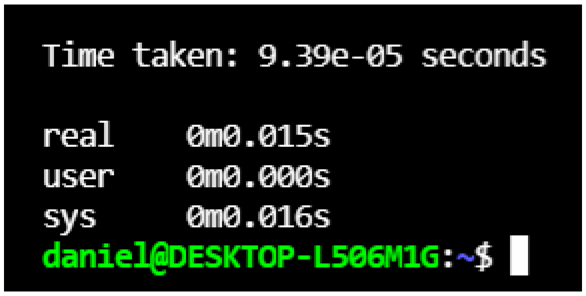
c) Code:



```
#include <iostream>
#include<ctime>
#define N 101

using namespace std;

const __int128 NIL = -1;
__int128 Fib_List[N];

void init()
{
    for(__int128 i=0; i<N; i++)
        Fib_List[i] = NIL;
}

__int128 fib(int x) {
    if(Fib_List[x] == NIL) {
        if(x <= 1)
            Fib_List[x] = x;
        else
            Fib_List[x] = fib(x-1) + fib(x-2);
    }
    return Fib_List[x];
}

std::ostream&
operator<<( std::ostream& dest, __int128_t value )
{
    std::ostream::sentry s( dest );
    if ( s ) {
        __uint128_t tmp = value < 0 ? -value : value;
        char buffer[ 256 ];
        char* d = std::end( buffer );
        do
        {
            -- d;
            *d = "0123456789"[ tmp % 10 ];
```



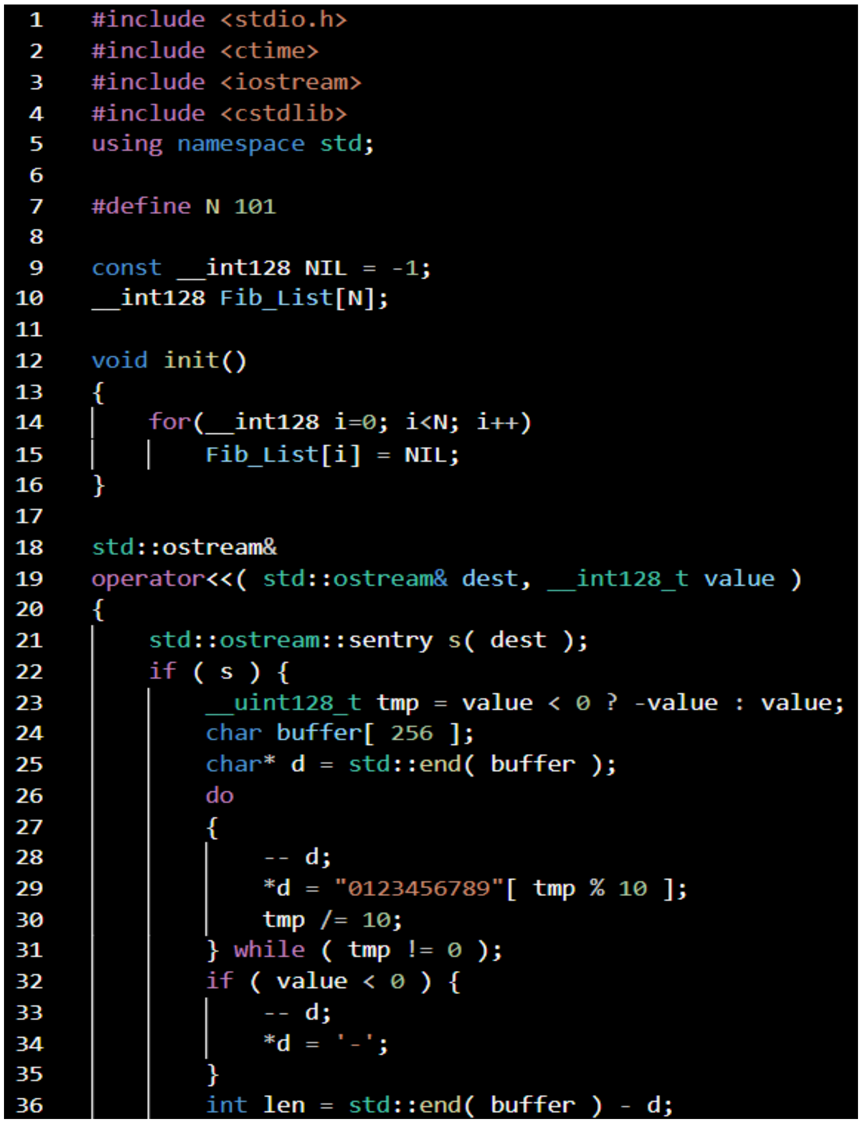
```
            tmp /= 10;
        } while ( tmp != 0 );
        if ( value < 0 ) {
            -- d;
            *d = '-';
        }
        int len = std::end( buffer ) - d;
        if ( dest.rdbuf()->sputn( d, len ) != len ) {
            dest.setstate( std::ios_base::badbit );
        }
    }
    return dest;
}

int main()
{
    init();
    timespec start, end;
    clock_gettime(CLOCK_REALTIME, &start);
    int n=99;
    cout << "\nFibonnaci number is : " << fib(n) << endl;

    clock_gettime(CLOCK_REALTIME, &end);
    long long secs = end.tv_sec - start.tv_sec;
    long long nanosecs = end.tv_nsec - start.tv_nsec;
    long double elapsed = secs + nanosecs*(long double)1e-9;
    cout<<"\nTime taken: "<<elapsed<<" seconds\n";
}
```



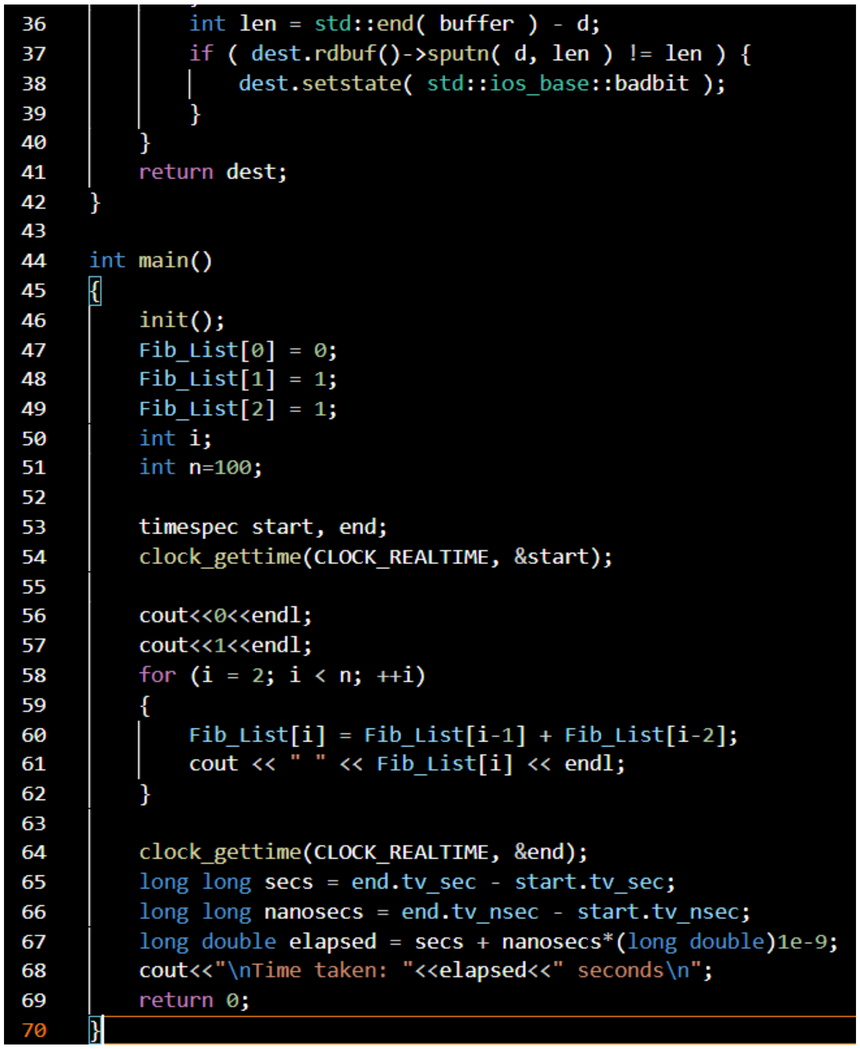
```
Time taken: 9.39e-05 seconds

real    0m0.015s
user    0m0.000s
sys     0m0.016s
daniel@DESKTOP-L506M1G:~$
```

Time Taken t(c) = **0.0000939 seconds.**
Speedup = t(a)/t(c) = **29,925.452609*(1.6180^57)**

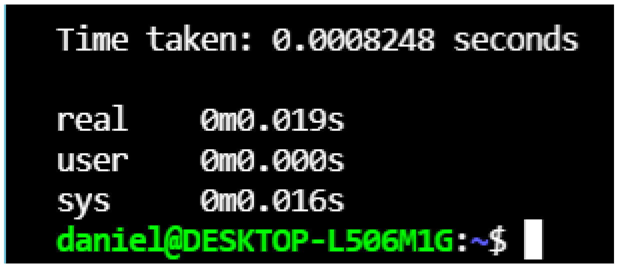
d) Code:



```
#include <stdio.h>
#include <ctime>
#include <iostream>
#include <cstdlib>
using namespace std;

#define N 101

const __int128 NIL = -1;
__int128 Fib_List[N];

void init()
{
    for(__int128 i=0; i<N; i++)
        Fib_List[i] = NIL;
}

std::ostream&
operator<<( std::ostream& dest, __int128_t value )
{
    std::ostream::sentry s( dest );
    if ( s ) {
        __uint128_t tmp = value < 0 ? -value : value;
        char buffer[ 256 ];
        char* d = std::end( buffer );
        do
        {
            -- d;
            *d = "0123456789"[ tmp % 10 ];
            tmp /= 10;
        } while ( tmp != 0 );
        if ( value < 0 ) {
            -- d;
            *d = '-';
        }
        int len = std::end( buffer ) - d;
```



```
        int len = std::end( buffer ) - d;
        if ( dest.rdbuf()->sputn( d, len ) != len ) {
            dest.setstate( std::ios_base::badbit );
        }
    }
    return dest;
}

int main()
{
    init();
    Fib_List[0] = 0;
    Fib_List[1] = 1;
    Fib_List[2] = 1;
    int i;
    int n=100;

    timespec start, end;
    clock_gettime(CLOCK_REALTIME, &start);

    cout<<0<<endl;
    cout<<1<<endl;
    for (i = 2; i < n; ++i)
    {
        Fib_List[i] = Fib_List[i-1] + Fib_List[i-2];
        cout << " " << Fib_List[i] << endl;
    }

    clock_gettime(CLOCK_REALTIME, &end);
    long long secs = end.tv_sec - start.tv_sec;
    long long nanosecs = end.tv_nsec - start.tv_nsec;
    long double elapsed = secs + nanosecs*(long double)1e-9;
    cout<<"\nTime taken: "<<elapsed<<" seconds\n";
    return 0;
}
```



```
Time taken: 0.0008248 seconds

real    0m0.019s
user    0m0.000s
sys     0m0.016s
daniel@DESKTOP-L506M1G:~$
```

Time Taken t(d) = **0.0008248 seconds.**

Speedup = t(a)/t(d) = **3,406.8865179*(1.6180^57)**

2) I have taken **C++** from Bucket-1 and **Python** from Bucket-2. In this question, I am just performing the matrix multiplication operations but not outputting it (I would have commented it).

a) CPU time = User time + Sys time

Bucket 1: C++

Matrix Size	System time (Double)	CPU time (Double)	System time (Integer)	CPU time (Integer)
32	0.016s	0.016s	0.016s	0.016s
64	0.016s	0.032s	0.016s	0.016s
128	0.031s	0.078s	0.016s	0.032s
256	0.016s	0.313s	0.016s	0.079s
512	0.016s	0.969s	0.016s	0.610s

Bucket 2: Python

Matrix Size	System time (Double)	CPU time (Double)	System time (Integer)	CPU time (Integer)
32	0.024s	0.10s	0.016s	0.160s
64	0.016s	0.061s	0.016s	0.067s
128	0.030s	0.153s	0.024s	0.172s
256	0.016s	0.275s	0.032s	0.246s
512	0.030s	0.742s	0.032s	0.721s

b) The total program execution time = User time + System time. This is because considering real time also takes into account the time delay due to user input.

Bucket 1: C++

Matrix Size	Meat time (Double)	Meat/Execution time (Double)	Meat time (Integer)	Meat/Execution time (Integer)
32	0.0005698s	0.0356125	0.0001105s	0.0069063
64	0.0069846s	0.2182688	0.000831s	0.0519375
128	0.0401501s	0.5147449	0.0076739s	0.2398094
256	0.277418s	0.8863195	0.0675223s	0.8547127
512	0.94904s	0.9794014	0.601782s	0.9865279

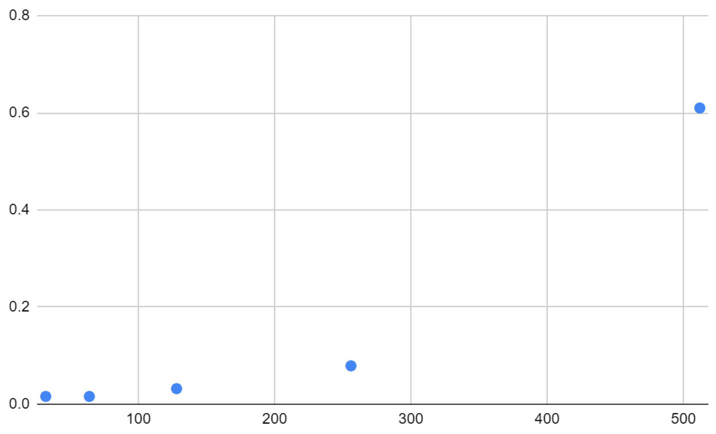
Bucket 2: Python

Matrix Size	Meat time (Double)	Meat/Execution time (Double)	Meat time (Integer)	Meat/Execution time (Integer)
32	0.0003743s	0.003743	0.0020535s	0.0128344
64	0.0002874s	0.0047114	0.0009731s	0.0145239
128	0.0009605s	0.0062777	0.0043766s	0.0254453
256	0.0018745s	0.0068164	0.0184298s	0.0749179
512	0.0070872s	0.0095514	0.2072755s	0.2874834

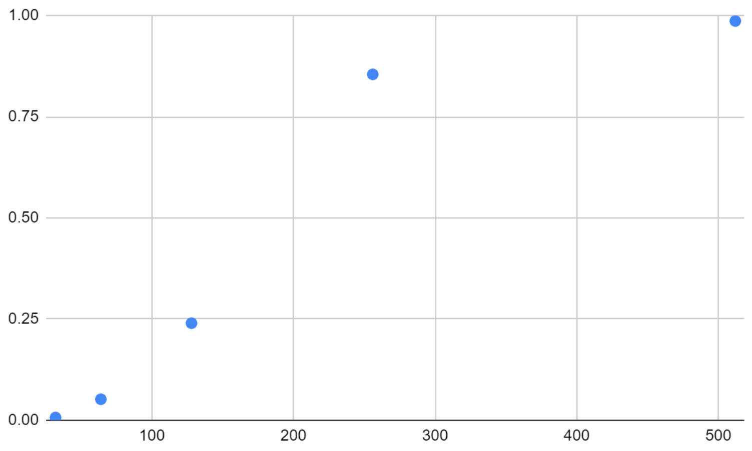
c) **Comparison between Bucket 1 (C++) and Bucket 2 (Python):**

Plots for C++:

1) CPU time(Integer) vs Matrix Size

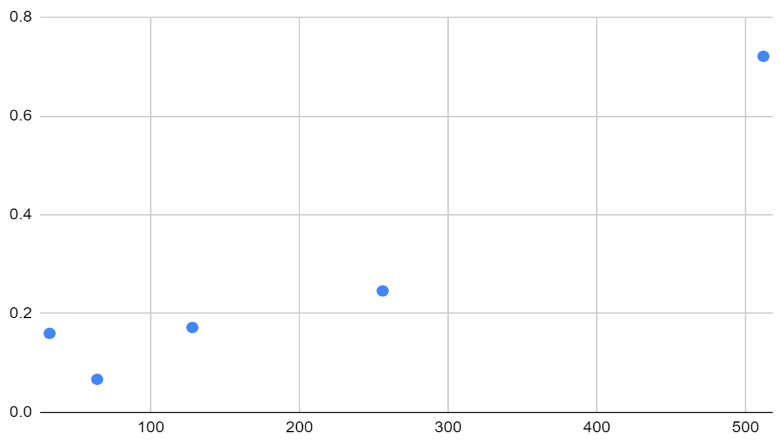


2) Meat/Execution time (Integer) vs Matrix Size

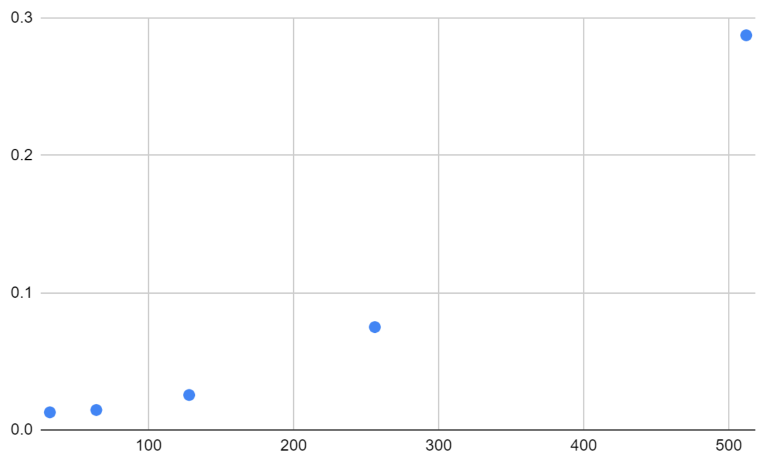


Plots for Python:

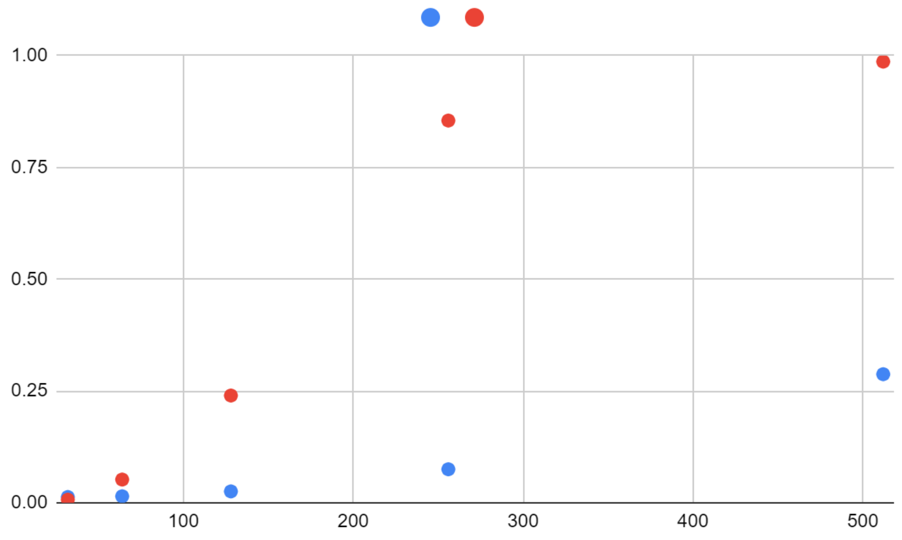
1) CPU time (Integer) vs Matrix Size



2) Meat/Execution time (Integer) vs Matrix Size:



Comparing C++ and Python's Meat/Execution time (Integer) vs Matrix Size:



Here, Red is C++
Blue is Python

Observations:

- The Meat/Execution time for C++ is much larger than in the case of Python.
- For larger values of N, the Meat/Execution time of C++ program almost equals 1.
- The execution time of the Python program is much faster than the C++ one.
- The execution time of C++ with Double datatype is much more than with the Integer datatype which is in contrary with Python as it has the vice-versa.

Name: Daniel Giftson E

Roll No: 20110051

Github link: https://github.com/Dany2002-hub/ES215_Assignment-1/tree/main