**Name: Sheetal More**

**Class: BE Comp II**

**Write a program non-recursive and recursive program to calculate Fibonacci numbers and analyze their time and space complexity.**

**Program:**

#include<iostream>

#include<vector>

using namespace std;

//Iteratively using memoization

int iStepFibbonacci(int n)

{

vector<int> f;

f.push\_back(0);

f.push\_back(1);

int cnt = 2;

for(int i = 2; i < n; i++){

cnt++;

f.push\_back(f[i - 1] + f[i - 2]);

}

return n;

}

int rSteps = 0;

int rStepFibbonacci(int n)

{

rSteps++;

if(n < 0) return 0;

if(n == 1 || n == 0) return 1;

return rStepFibbonacci(n - 1) + rStepFibbonacci(n - 2);

}

int main()

{

int n;

cin >> n;

cout << "Fibbonacci Value : " << rStepFibbonacci(n) << '\n';

cout << "Steps required using Iteration : " << iStepFibbonacci(n) << '\n';

cout << "Steps required using recursion : " << rSteps << '\n';

return 0;

}

**Output:**

