**DAA Assignment 1**

**Analysis of Recursive and Iterative Programs**

**Neeti Kurulkar**

**#include <iostream>**

**#include <ctime>**

**#include <iomanip>**

**using namespace std;**

**// Iterative Fibonacci**

**int fib\_iterative(int n) {**

**if (n <= 1) return n;**

**int a = 0, b = 1, c;**

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

**c = a + b;**

**a = b;**

**b = c;**

**}**

**return b;**

**}**

**// Recursive Fibonacci**

**int fib\_recursive(int n) {**

**if (n <= 1) return n;**

**return fib\_recursive(n - 1) + fib\_recursive(n - 2);**

**}**

**int main() {**

**int n;**

**cout << "Enter a number: ";**

**cin >> n;**

**// Iterative**

**clock\_t start\_iter = clock();**

**int result\_iter = fib\_iterative(n);**

**clock\_t end\_iter = clock();**

**double duration\_iter = double(end\_iter - start\_iter) / CLOCKS\_PER\_SEC;**

**int space\_iter = sizeof(n) + sizeof(int) \* 3;**

**// Recursive**

**clock\_t start\_rec = clock();**

**int result\_rec = fib\_recursive(n);**

**clock\_t end\_rec = clock();**

**double duration\_rec = double(end\_rec - start\_rec) / CLOCKS\_PER\_SEC;**

**int space\_rec = sizeof(n) \* (n > 1 ? n : 1);**

**// Output**

**cout << "\n------------------------------------------\n";**

**cout << "\n[ Iterative Approach ]\n";**

**cout << "Fibonacci(" << n << ") : " << result\_iter << endl;**

**cout << "Time required : " << fixed << setprecision(6) << duration\_iter << " seconds" << endl;**

**cout << "Space required : " << space\_iter << " bytes" << endl;**

**cout << "\n[ Recursive Approach ]\n";**

**cout << "Fibonacci(" << n << ") : " << result\_rec << endl;**

**cout << "Time required : " << fixed << setprecision(6) << duration\_rec << " seconds" << endl;**

**cout << "Space required : " << space\_rec << " bytes" << endl;**

**cout << "------------------------------------------\n";**

**return 0;**

**}**

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**