
Programming Project #1 [100 points].

Due date: Thursday, February 20.

(A) [15 points] Write a recursive function that will calculate Fibonacci numbers using a recursive definition. Write a short program to test it. The input of this program must be a positive integer N; the output is the corresponding Fibonacci number F_N.

(B) [20 points] Write an iterative function to calculate Fibonacci numbers. Write a test driver for it. The input of this program must be a positive integer N; the output is the corresponding Fibonacci number F_N.

(C) [10 points] Write a program that will compare running time of the recursive and iterative functions for calculating Fibonacci numbers. Call each function for the same size of input N (the index of a Fibonacci number to be calculated) and find their running times. For part (E) of this project you will have to run this program multiple times to find out how the running time of each function depends on the value of N.

(D) [35 points] In this part of the project, write a function that will implement each Fibonacci number with the help of an integer array of size 100 (elements of this array will be digits of the Fibonacci number). When the function is called to find F_N, it will calculate all Fibonacci numbers from F_2 to F_N using the basic formula $F_N = F_{N-1} + F_{N-2}$. To add two Fibonacci numbers, the function will add elements of two arrays corresponding to F_{N-1} and F_{N-2} and store their sums in the array corresponding to F_N. (You are NOT supposed to store ALL arrays.) Write a program that will calculate Fibonacci numbers containing up to 100 digits. Display the biggest Fibonacci number that has less than 100 digits. Make sure that your program will display the error message when a Fibonacci number has 100 digits or more.

(E) [20 points] This part of the project is a written analysis of two algorithms of calculating Fibonacci numbers: recursive (part A) and iterative (part B). Show the theoretical order of growth of the running time for both algorithms. Then include experimental results from part (C) and explain them.

Submit the following:

→ files for programs (A) – (D), each file containing the code of ONE program (as a TEXT) and results of its execution (screenshots or snippets).

→ analysis for the part (E), which MUST be typed. The analysis may be submitted electronically or printed and submitted in the class. The project will not be graded until ALL the parts are submitted.

→ early submission is counted by submission of files for parts (A) – (D), which means that part (E) should be submitted not later than the dead line in order to get credit for early submission.