

M03S03 [Recursive Functions and Pointers]: Exercise 03

CS110: Computing Lab
Department of CSE, IIT Guwahati
Jan-May 2018

Problem Description

A recursive function is a function that calls itself. Every recursive function comprises of two essential steps known as base case and recursive step. At some point, the recursive function must encounter a subtask that it can perform without calling itself. This case, where the function does not recur, is called the base case. A recursive method solves a problem by calling a copy of itself to work on a smaller problem. This step is called recursion step.

The Fibonacci sequence is defined as follows:

$$F_1 = 0 \tag{1}$$

$$F_2 = 1 \tag{2}$$

$$F_{2n} = F_{n+1}^2 + F_n^2 \tag{3}$$

$$F_{2n+1} = 2F_n F_{n+1} + F_{n+1}^2 \tag{4}$$

Write a program to compute the k^{th} Fibonacci number using recursion.

Your program must take k as an input from user during runtime. Avoid redundant computation while finding F_k . That is, you have to compute $F_m, \forall m < k$ only once. (Hint: You can use an array to store the values of $F_m, \forall m < k$.)

Example 1:

k: 15

Expected Output: 987

Example 2:

k: 25

Expected Output: 121393

Example 1:

k: 6

Expected Output: 13