Object-Oriented-Programming Coursework 5 Report

20191339 chanhwiHwang

In this coursework, I implemented two programs, the first program is to receive an integer through file stream, calculate the binary base number, and then write it to output file. And second program is to receive an integer n between 1 and 30, and calculate the n-th term of Fibonacci sequence. In both program, I used recursive function to conveniently calculate the result. In first program, I defined myIntegerToBinary func, which receives integer n and returns binary converted string. Since the method of converting an decimal to binary is to repeatedly divide it with 2 and write each remainder in inverse order, I first defined base case - when n is less than 2, which is not divisible by 2, it returns itself. And in recursive case, it calls itself with n / 2 as argument so that it can receive next call's return value, append current remainder of n divided by 2 to it and return it to previous call. So it can obtain the consecutive series remainders of n divided by 2 in inverse order. In the second program, I defined myFibonacci func, which receives integer n and returns n-th Fibonacci number. In this function, I defined an array to store the return value of myFibonacci func, since recursive call of previous arguments can cause repetitive call of func with same argument, which leads to unnecessary recursive call. So, it first checks if n is 0 or 1, which is base cases, and returns 0 and 1 at each case. If n is not the base case, check if the array's corresponding index, n - 2's index since maximum value of n is 30 and 0, 1 are excluded from it, exists. If the value exists, return it and if not, store the sum of return values of myFibonacci(n -1) and myFibonacci(n-2) to corresponding index of array and return it.