In Part 2 of the assignment you will analyze two algorithms, written in Java-like pseudocode, for calculating powers of real numbers:

- SP(x,n) // SP stands for "slow power"
- FP(x,n) // FP stands for "fast power"

These algorithms both take two parameters: a non-zero real number called x, and a natural number called n. The algorithms do not perform any error checking of the values passed to them. They are simply assumed to be correct. These algorithms both return a real number which is xn

Here is pseudocode for the algorithms:

SP(x,n)	FP(x,n)
result = 1 for (i=1; i<=n; i++) result = result *x return result	if n==0 return 1 if n is even return FP(x*x, n/2) else return x*FP(x*x,(n-1)/2)

Q1 Preliminary Analysis (7 marks)

In this analysis you will be comparing the efficiency of the two alorithms by counting the use of the most expensive operation in the algorithms, which is multiplication. (Note that because all the divisions in FP are integer divisions by 2, they can be implemented efficiently as right binary shifts).
In the questions that follow, it doesn't matter what x is.

a) (1 mark) How many multiplications will be used in total to calculate SP(x,n)?

b) (3 marks) Trace the execution of the return statements of FP(x,53), printing them statements one by one as they are being <u>called</u> and stating next to it the number of multiplications in that statement. The first line of the answer is:

    return x*FP(x*x,26)   // 2 multiplications (in red: the first one, in the first parameter, is executed before FP is called. The second one is executed after)

c) (2 marks) Add one line to the FP algorithm to reduce the total number of multiplications in FP(x,n) when FP is initially called with n > 0. Rewrite the algorithm underneath:

d) (1 mark) What is the total number of multiplications used by your new algorithm to calculate FP(x,53)?

Q2 – Proof of Correctness (21 marks)

In this question you will use strong induction to prove that your new algorithm works correctly.
In other words, you will prove that $\forall n \in \mathbb{N}\ \forall x \in \mathbb{R}-\{0\}\ FP(x,n) = x^n$

a) Predicate Function (1 mark)

Your conjecture has already been stated in symbolic form:
It is a statement of the form $\forall n \in \mathbb{N}$, P(n)
What is the predicate function P(n)?

b) Proof: Base cases (4 marks)

c) Proof: Inductive step setup (2 marks)

This is the beginning of the inductive step where you are stating the assumptions in the inductive step and what you will be proving in that step. As you do so, identify the inductive hypothesis.

d) Proof: Inductive step (14 marks)

Q3 – Proof of Cost (22 marks)

In this question you will use strong induction to prove that your new algorithm is very efficient.
Given a non-zero real number x, and a natural number n, define CFP(x,n) to be the cost of FP(x,n) = the total number of multiplications in the total execution of FP(x,n)
You will prove that $\forall n \in \mathbb{N}^+ \forall x \in \mathbb{R}\text{-}\{0\}\ \mathrm{CFP}(x,n) \leq 2\log_2 n$

a) Predicate function (1 mark)

Your conjecture has already been stated in symbolic form:
It is a statement of the form $\forall n \in \mathbb{N}^+$, P(n)
What is the predicate function P(n)?

b) Proof: Base cases (2 marks)

c) Proof: Inductive step setup (2 marks)

This is the beginning of the inductive step where you are stating the assumptions in the inductive step and what you will be proving in that step. As you do so, identify the inductive hypothesis.

d) Proof: Inductive step (17 marks)