D1-2

Mahshad Saghaleini

July 30, 2019

team repository name : SEPCalculatorTeamI
Personal repository name : Soen-6011

Chapter 1

Problem 1

1.1 Introduction

This is an exponential function, The exponent is a variable rather than a constant and its base is represented with constant rather than variable. The output of this function can grow very fast. Suppose f(x) = ab^x, “b” is a change factor which can be constant , “a” is the initial value of the function and “x” is independent variable or the output of the function. Exponential function has the general form y = f(x) = ab^x where $ab > 0$ and $ab" \neq "1$ and “x” can be any real number. The reason $a > 0$ is that if it is negative, the function is undefined for $-1 < x < 1$. Restricting a to positive values allows the function to have a domain of all real numbers. The domain of a function is the specific set of values that the independent variable in a function can take on. The domain of exponential functions is all real numbers. The range is all real numbers greater than zero. The line $y = 0$ is a horizontal asymptotic for all exponential functions. When $a > 1$ as x increases, the exponential function increases, and as x decreases, the function decreases. On the other hand, when $0 < a < 1$ as x increases, the function decreases, and as x decreases, the function increases. The exponential function is very useful and relevant for modeling the behavior of systems whose relative growth rate is constant for example bank accounts and populations. The natural exponential function y = e^x which is defined for all x $\in R$ and whose range is (0,∞).If x is a rational number because any irrational number can be expressed as the limit of a sequence of rational numbers Exponentiation of a positive real number b with an arbitrary real exponent x can be defined b by continuity with the rule:

b$^x = \lim b \to Rb^r$

where the limit as r gets close to x is taken only over rational values of r, this limit only exists for positive b. If b is an algebraic number different from 0 and 1, and x an irrational algebraic number, then all values of bx (there are infinitely many) are transcendental.

Chapter 2

Problem 2

2.1 Assumption

String and character is not acceptable as an input for the function. to limit the growth of the function we assume 'x' is not a big real number. We assume that the user will not give a number less or equal to zero as an input to the base b. since the domain of x as the exponent of the function is all real number I assume that it has rational and irrational number as input.

2.2 Requirement

2.2.1 F6.1

The domain of "a" is all the real number greater then '0'. If a is equal to 0, then no mater what is 'b' and 'x' the out put will be '0' and if a is less than '0' the function when $-1 < x < 1$ is undefined. version number 1.3

2.2.2 F6.2

The domain of "b" is all the real number greater than '0', if b is less or equal to '0' then f(x) is not defined. version number 1.3

2.2.3 F6.3

The domain of "x" is all the real number.If x is equal to '0', the output is depend on a. version number 1.3

2.2.4 F6.4

since the domain of x as the exponent of the function is all real number we

Chapter 3

Problem 3

3.1 Algorithm

3.1.1 Algorithm 1:

Algorithm 1:

Calculating the value of ab^x from the exponential function

1. class ab^x UsingExponentialFunction
2. Function b^x(Argument x ,Argument b,Power)
3. for (int i=0 ; i less than x ; i++)
4. power of base is equal to 0
5. b is equal to b times b
6. power of base is equal to new b
7. return power of base
8. end of function
9. function calculate the result
10. Calculate a times to power of base
11. return result
12. end
13. In Main Function
14. Take the input value of a from the UI provided by the user
15. create an int data type initialValue and assign a to it
16. Take the input value of b from the UI provided by the user
17. create an int data type commonRatio and assign b to it
18. Take the input value of x from the UI provided by the user
19. create a double data type exponentValue and assign x to it
20. Send the exponentValue and commonRatio to the power function
21. Send the initialValue and result of the power function to the calculate function
22. print the value of ab^x to the console application
23. end main function

3.1.2 Algorithm 2:

Algorithm 2:

Calculating the value of ab^x from the exponential function

1. class ab^x UsingExponentialFunction
2. Function b^x(Argument x ,Argument b,Power)
3. check if x is even
4. power is equal to
5. b to the power of 2
6. to the power of x divided by 2
7. else
8. if x is odd
9. power is equal to b multiple to the b to the power of 2
10. to the power of x minus 1 divided by 2
11. return the result
12. function calculate the result
13. Calculate a times to power of base
14. return result
15. end
16. In Main Function
17. Take the input value of a from the UI provided by the user
18. create an int data type initialValue and assign a to it
19. Take the input value of b from the UI provided by the user
20. create an int data type commonRatio and assign b to it
21. Take the input value of x from the UI provided by the user
22. create a double data type exponentValue and assign x to it
23. Send the exponentValue and commonRatio to the power function
24. Send the initialValue and result of the power function to the calculate function
25. print the value of ab^x to the console application
26. end main function

3.2 Algorithm

3.2.1 technical reasons

For the second algorithm, When x is even, the value at the next level is exactly half; when x is odd, the value is a little less.On the other word each time we go one level deeper in the recursion, the value of x at the new level is at most half of what it was. This will always be true by looking at our definition of exponential. Time complexity (big O) for the recursion function is linear and so it looks more efficient.As a reason of efficiency I chose the second algorithm for my function implementation.Also