Problem 2

The assumptions and requirements for the function

$$x^y$$

as per ISO/IEC/IEEE 29148 standards.

2.1 Assumptions

- Assumption 1
  - ID: ASSUMP1
  - Version: 1.0
  - Type: functional
  - Owner: Pavit Srivatsan
  - PRIORITY: 1
  - Difficulty: Easy
  - DESC: fractional inputs are entered as double values
  - Rationale: when input base or exponent value equals 2/3, it must be expressed as 0.67
- Assumption 2
  - ID: ASSUMP2
  - Version: 1.0
  - Type: functional
  - Owner: Pavit Srivatsan
  - PRIORITY: 1
  - Difficulty: medium
  - DESC: The output of larger values of exponent and base are expressed in terms of exponents
  - Rationale: when input is 100 raised to 100, the output is expressed as 1.9047931533522278E25
- Assumption 3
  - ID: ASSUMP3
  - Version: 1.0
  - Type: functional
  - Owner: Pavit Srivatsan
  - PRIORITY: 2
  - Difficulty: High
  - DESC: Users enter whole numbers and rational numbers
  - Rationale: Irrational numbers are not handled by the code.For Example: π, $\sqrt{2}$
- Assumption 4
  - ID: ASSUMP4
  - Version: 1.0
  - Type: functional

- Owner: Pavit Srivatsan
- PRIORITY: 3
- Difficulty: Easy
- DESC: Mathematical symbols such as infinity, indeterminate are represented in words
- Rationale: Symbols such ∞ are represented in words - infinity

2.2 Functional Requirements

- Requirement 1
  - ID: FUNR1
  - Version: 1.0
  - Type: functional
  - Owner: Pavit Srivatsan
  - PRIORITY: 1
  - Difficulty: Easy
  - DESC: The arguments passed to the function x power y shall be valued within $-\propto to+\propto$ and fractions are expressed as double values.
  - Rationale:when $x = 2.2$, $y = 4.45$ where x expressed in radians.
- Requirement 2
  - ID: FUNR2
  - Version: 1.0
  - Type: functional
  - Owner: Pavit Srivatsan
  - PRIORITY: 2
  - Difficulty: Easy
  - DESC: The function shall return the value 1 when any base value is raised to the power zero
  - Rationale: 10 raised to the power 0 returns 1
- Requirement 3
  - ID: FUNR3
  - Version: 1.0
  - Type: functional
  - Owner: Pavit Srivatsan
  - PRIORITY: 2
  - Difficulty: Easy
  - DESC: The function shall return zero when base value zero is raised to any exponent value
  - Rationale: 0 raised to the power 10 returns 0
- Requirement 4
  - ID: FUNR4
  - Version: 1.0

- Type: functional
- Owner: Pavit Srivatsan
- PRIORITY: 2
- Difficulty: Medium
- DESC: The function shall accept only numerical values as specified in the domain
- Rationale: string values, numbers with special characters, special characters are not allowed as inputs and an appropriate error message is displayed

Problem 3

Pseudocode and Algorithm

Calculate: $f(x, y) = x^y$

ALGORITHM 1: Iterative algorithm to calculate x^y

1. function **power_function_iterative(x,y)**
in: double number x,y
out: double number result
2. $result \leftarrow 1$
3. $temp \leftarrow 1$
4. for $temp \leq y$ do
5. $\quad result \leftarrow result * x$
6. $\quad temp \leftarrow temp + 1$
7. end for
8. return result

The output is stored in result, which is initially set to 1. It is then looped from 1 to y, x number of times, incremented by one on each iteration and on each iteration we multiply result by x. At the end of the loop value of result is equal to x^y.

ALGORITHM 2: Recursive Divide and Conquer algorithm to calculate x^y

1. function **power_function_recursive(x,y)**
in: double number x,y
out: double number result
3. $power \leftarrow exponent\_helper(x, y)$
4. $result = power$
5. return result

---

1. function **exponent_helper(x,y)**
in: double number x, y
out: double number sum
2. if $x < 0$ then
3. $\quad x \leftarrow 1.0/x$
4. $\quad y \leftarrow -y$
5. $\quad$ return $exponent\_helper(x, y)$
6. else if $y = 0$ then
7. $\quad$ return 1.0
8. else if $y = 1$ then
9. $\quad$ return x
10. else if $y \mod 2 = 0$ then
11. $\quad y \leftarrow y * y$
12. $\quad y \leftarrow y/2$
13. $\quad$ return $exponent\_helper(x, y)$
14. else
15. $\quad x \leftarrow x * x$
16. $\quad y \leftarrow y - 1$
17. $\quad y \leftarrow y/2$
18. $\quad$ return $exponent\_helper(x, y)$
19. end if

A helper function called exponent_helper is defined which calculates x^y. In the base case when $y = 0$, we return 1, otherwise when $y = 1$ we return x. When x is even we recurse on $x = x * x$ and $y = y/2$. In case when x is odd we recurse on $x = x * x$ and $y = (y - 1)/2$. In the end, in our main function power_function_recursive we multiply the result of exponent_helper to the value of a and return our result.

Advantages and Disadvantages

Algorithm 1:

Advantages:
1. In terms of space complexity, iterative algorithms don't suffer from stack overflow because all operations are done on the heap.
2. They are easy to comprehend by humans and have better readability.
Disadvantage:
1. The time complexity of the iterative algorithm is $O(n)$, hence it is not very efficient for larger inputs in terms of time.
2. Proper terminating condition for loop is important or else it might lead to infinite looping.

Algorithm 2:

Advantages:
1. The time complexity of the recursive algorithm is $O(logn)$. This recursive algorithm is optimized (tail recursive) so that we don't get stack overflow error and it handles large inputs better.
2. Recursion has higher maintainability than looping. Handling the base case properly requires little or no modifications.
Disadvantages:
1. Due to the continuous allocation of memory space leading to a stack overflow, efficiency is significantly affected.
2. It is difficult to comprehend. A certain level of expertise is required for understanding it vividly.

Conclusion

Recursive algorithm which is based on Divide and Conquer Algorithmic Strategy is preferred over iterative algorithm

References

[1] TutorialsPoint,
`https://www.tutorialspoint.com/java/lang/math_pow.htm`

[2] GeeksforGeeks,
`https://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/`

[3] MathBitsNotebook,
`https://mathbitsnotebook.com/Algebra1/FunctionGraphs/FNGTypeExponential.html`