1. Stack

a. Explain what you know about Stack and give 3 examples of its use!

b. Explain the implementation of a Stack using arrays that you understand!

Infix Notation: (6 - 3) * (1 + 8) – 2 * (4 + 2)

c. **Convert** that infix notation to **Postfix Notation**. Simulate it using **Stack Algorithm**.

d. **Evaluate** the **Postfix Notation** from before. Simulate it using **Stack Algorithm** so the **calculation result** can be achieved.

e. **Convert** that infix notation to **Prefix Notation**. Simulate it using **Stack Algorithm**.

f. **Evaluate** the **Prefix Notation** from before. Simulate it using **Stack Algorithm** so the **calculation result** can be achieved.

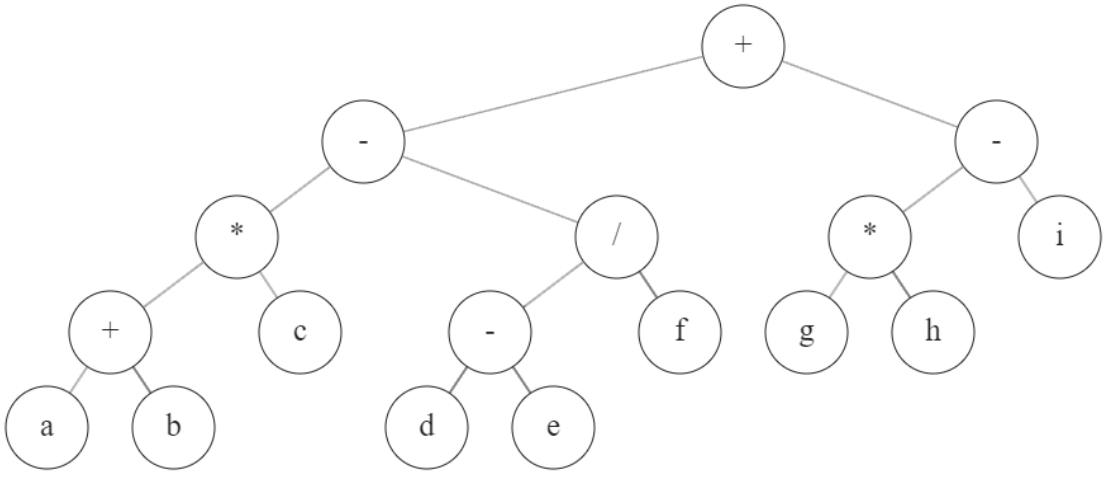
2. Expression Tree

a. **Create** the **Expression Tree** from this inorder expression (you can use paint and paste the image in Ms.Word): **4 – (2 * 5) + (3 – (1 * 3) + 5)**

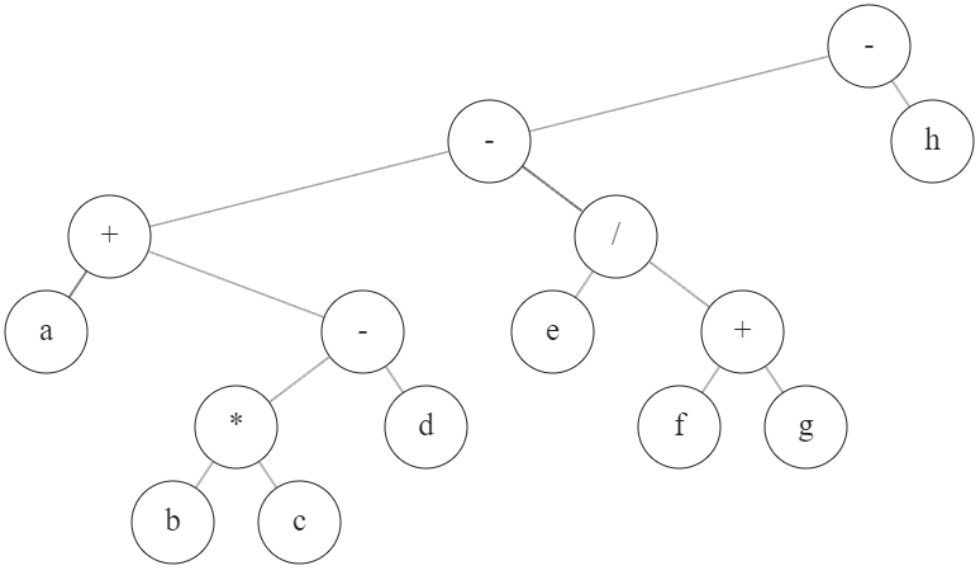
b. **Create** the **Expression Tree** from this preorder expression (you can use paint and paste the image in Ms.Word): ***3 – +*6 2 1 – 2 +2 4**

c. **Create** the **Expression Tree** from this postorder expression (you can use paint and paste the image in Ms.Word): **7 4 3 * 1 5 + / -**

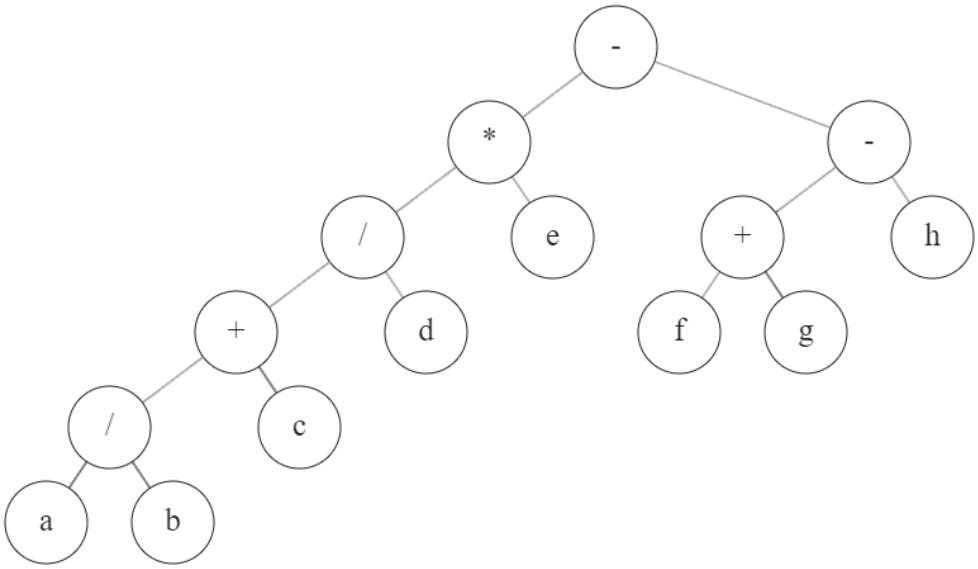
d. **Create** the **Inorder** from the Expression Tree below:



e. **Create** the **Preorder** from the Expression Tree below:



f. **Create** the **Postorder** from the Expression Tree below:



3. Hashing & Hash table

Tropical Delight, a prominent fruit distributor in the Grand Line, is looking to improve their data management system using a hash table. They have hired you as a programmer to design their data storage system. During your meeting with the company's IT team, the head of the division asks you to create a hash table with the following criteria:

The hash function used to enter data in the hash table involves 3 stages:
Stage 1: Multiply all decimal values of characters.
Stage 2: Extract the last 3 digits of the number.
Stage 3: Perform modulo operation with the table size.

To handle data collision, the company decides to use open addressing with **linear probing**. You are asked to simulate the storage of data with the given hash function criteria to produce an index in the hash table.

Notes:

- The maximum length of the hash table is 200
- For reference, the character 'a' has a decimal value of 97 in ASCII table

The following are the key-value pairs that must be entered in the hash table:
pineapple-45
papaya-32
banana-12
coconut-67

Your task is to create a simulation of how these key-value pairs would be stored in the hash table according to the specified hash function and collision handling technique. Here's an ASCII table to help. Good luck!