Problem 1 – Job Requests

In the futuristic city of Byteburg, the government maintains a Digital Archive System to store

classified records efficiently. The system follows a B-tree of order 3 to keep the records

structured, ensuring fast insertion, retrieval, and balancing.

However, a recent system failure corrupted the archive, and Chief Engineer Nova needs your

help to restore its functionality.

Your task is to implement the Digital Archive System with the following capabilities:

1. Add a record (insert a key) – Each record is assigned a unique numeric ID (a positive

integer). You must insert these IDs into the archive while maintaining the B-tree

structure.

2. Display the archive structure – Show how records are organized in the B-tree using an

indented format to represent levels in the tree.

3. Search for a record – Given a record ID, determine whether it exists in the archive and

print an appropriate message.

A B-tree of order 3 (B(3) tree) is a balanced search tree with the following properties:

● Each node can store a maximum of 2 keys (since order = 3).

● Each node can have at most 3 children.

● The tree remains balanced—all leaf nodes are at the same level.

● When a node exceeds its limit, it splits, pushing a key to the parent node.

Input Format- An integer N – The number of records to be inserted.

N space-separated integers – The record IDs to insert in order

An integer Q – The record ID to search.

Output Format- N space-separated integers – The record IDs to inserted in order

Boolean true or false - If search ID is found

Problem 1 – Job Requests

In the futuristic city of Byteburg, the government maintains a Digital Archive System to store

classified records efficiently. The system follows a B-tree of order 3 to keep the records

structured, ensuring fast insertion, retrieval, and balancing.

However, a recent system failure corrupted the archive, and Chief Engineer Nova needs your

help to restore its functionality.

Your task is to implement the Digital Archive System with the following capabilities:

1. Add a record (insert a key) – Each record is assigned a unique numeric ID (a positive

integer). You must insert these IDs into the archive while maintaining the B-tree

structure.

2. Display the archive structure – Show how records are organized in the B-tree using an

indented format to represent levels in the tree.

3. Search for a record – Given a record ID, determine whether it exists in the archive and

print an appropriate message.

A B-tree of order 3 (B(3) tree) is a balanced search tree with the following properties:

● Each node can store a maximum of 2 keys (since order = 3).

● Each node can have at most 3 children.

● The tree remains balanced—all leaf nodes are at the same level.

● When a node exceeds its limit, it splits, pushing a key to the parent node.

Input Format- An integer N – The number of records to be inserted.

N space-separated integers – The record IDs to insert in order

An integer Q – The record ID to search.

Output Format- N space-separated integers – The record IDs to inserted in order

Boolean true or false - If search ID is found