# **Tutorial 7**

# **Hash Table**

This task requires you to study and modify the programs provided to solve collision problems in hash table:

* Linear probing collision
* Double hashing

Study the codes from the GitHub repository:

<https://github.com/zairulmazwan/Tutorial7.git>

The repository consists of:

* Collision\_LinearProbing.java
* Collision\_DoubleHashing.java
* ReadFile.java

Apply the program for the dataset provided (**Appendix A**).

The dataset consists of four fields/variables i.e ***CountryName***, ***Alpha-2Code***, ***Alpha-3Code***, ***NumbericCode***, and ***ISO***. There are 242 records.

Table

Description automatically generated

Use ***NumericCode*** as the key to store ***CountryName*** in a 2D array of size 242.

Answer the following questions:

1. Which country is stored in the first position of the array?
2. Which country is stored in the last position of the array?
3. Which country is stored in the 53rd position of the array?
4. What is United Kingdom position in the array?

# **Doubly Linked List**

From the hash table tutorial, extend your work by using chaining approach in solving collisions in a hashtable. Use the dataset from ***iso-country-codes.csv*** to store ***CountryName*** into an array of size **100**.

Study the program from the repository:

<https://github.com/zairulmazwan/Tutorial7_DoublyLinkedList.git>

The repository consists of:

* DoublyLinkedList.java

Where the class consists of Node class, and methods needed in managing doubly linked list.

* Collision.java

This class demonstrates application of the doubly linked list class for a small

scale dataset.

Answer the following questions:

1. How many nulls in the table after all countries have been stored using chaining approach?
2. Which address of the array has the elements the most? What are those countries? Write a method to get this.
3. Based on the *findNode* method, write a method that returns the data of a key that being searched.

**Appendix A**

