**Lab 3 Discrete**

**Sets and Bit manipulation**

Ahmed Ragy 22011690

John William 22010701

**Problem Statement**

There is a need for an efficient and scalable solution to perform essential set operations such as union, intersection, complement, difference, and cardinality. The implementation leverages bitwise operations for high-performance computations, representing sets as binary strings where each bit corresponds to an element in a universal set.

This solution aims to minimize computational overhead and provide an efficient tool for performing complex set operations.

**Used Data Structures**

* ArrayList (List<Set> and List<String>):
* HashMap (Map<String, Integer>):
* String Array (String[])

**Assumptions**

* It is assumed that the elements in the universal set are unique and distinct to avoid ambiguity in indexing and bitwise operations.
* All subsets provided by the user are assumed to be in the universal set.
* For operations involving two subsets (e.g., union, intersection, and difference), it is assumed the user selects valid subset indices.
* The implementation assumes case sensitivity for element names. For example, "Apple" and "apple" are treated as different elements.
* Subsets are referenced by integers preceded by the letter "S" (e.g., S1, S2). It is assumed that users will follow this naming convention when performing operations.
* The maximum size of the universal set is constrained by the size of the data type used for bitwise operations (e.g., 32 bits for integers).
* Subset names (e.g., S1, S2, ...) are assumed to be sequentially generated, starting from 1, to ensure consistent indexing.
* Error Handling for Invalid Operations.
* End Keyword (case-insensitive) terminate input for both the universal set
* The program assumes that the universal set is non-empty.

**Sample runs**

**Test Case 1**

A screenshot of a computer program

Description automatically generated

**Test Case 2**

A screenshot of a computer

Description automatically generated

**Test Case 3**

A screenshot of a computer program

Description automatically generated

**Test Case 4**

A screenshot of a computer program

Description automatically generated

**Test Case 5**

A screenshot of a computer program

Description automatically generated

**Test Case 6**

A screenshot of a computer program

Description automatically generated