



## Lab 1 Sets and Bits Manipulation

### 1 Part 1: Basic Bit Operations

You have to implement 4 bits operations, so your program might allow user to choose one of the following operations.

1. **getBit(int number, int position):** This function returns the bit value (an integer, 0 or 1) in the number at position *position*, according to its binary representation. The least significant bit in a number is position 0.
2. **setBit(int number, int position):** This function set the bit value ( to be 1) in the number at position *position*, according to its binary representation. The least significant bit in a number is position 0 and return number after setting the bit.
3. **clearBit(int number, int position):** This function cleat the bit value ( to be 0) in the number at position *position*, according to its binary representation. The least significant bit in a number is position 0 and return number after clearing the bit.
4. **updateBit(int number, int position, boolean value):** This function set the bit value according to value parameter which is false (0) or true (1) in the number at position *position*, according to its binary representation. The least significant bit in a number is position 0 and return number after update.

### 2 Part 2: Sets Operations using Bits Manipulation

1. Implement a Set data structure that takes in the constructor a **list of strings as a Universe (U)**. The elements in the Set are subset of U. **You must use bits to represent the set.** The Set data structure should include the main operations:
  - 1) Add string to the set
  - 2) Union with another set
  - 3) Intersection with another set
  - 4) Complement of the set
  - 5) Difference from another set
  - 6) Cardinality of the set
  - 7) Get elements of the set



2. Write a program that.

- (a) Asks the user to enter a list of strings as a Universe (U)
- (b) Then asks for a number of sets (that are subsets of U). The user will enter the elements in each set
- (c) Then asks the user about the operations they want to perform:
  - 1) Union of two sets
  - 2) Intersection of two sets
  - 3) Complement of a set
  - 4) Difference between two sets
  - 5) Cardinality of a set
  - 6) Print a set

### 3 Part 3: Applications for Bits Manipulation

1. Write a function that takes a non-empty array of integers *nums*, where every element appears twice except for one integer, and returns the unique integer. You must implement a solution with a linear runtime complexity and use only constant extra space. you must think for your solution using bits manipulation operation.
  - (a) **[Bonus]** Assume there are two unique integers in the array. Implement a function that prints these two unique integers. You must solve it using bitwise operations.
2. Write a function that takes an unsigned integer and returns the number of '1' bits it.

### 4 Submission

- You must work **individually** and use **Java programming language** in your implementation.
- You are **not allowed** to use built in Set data structure or any data structure similar to it.
- Make sure you provide a clear and detailed report. It should contain:
  1. Problem statement.
  2. Used data structures.
  3. Algorithms documented using flow charts or pseudo code.
  4. Code Snippets.
  5. Sample runs and different test cases.
  6. Assumptions and details you find necessary to be clarified.

**Good Luck,,,**