**Name:** Sara Mahmoud Mahmoud ID:20010652

**Name:** Nisreen Hisham Fathi ID: 20012110

Sets and Bits Manipulation-Report

Report content:

1. Problem statement.
2. Used data structures.
3. Algorithms used 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

* **Problem statement:**

Part 1:Using Basic Bit Operations to do :

1. getBit(int number, int position):This function returns the bit value (an integer, 0 or 1) in the number at 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, 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 clear the bit value ( to be 0) in the number at 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(an integer, 0 (false) or 1(true) in the number at position, according to its binary representation. The least significant bit in a number is position 0 and return number after update.

Part 2: sets operations using Bits manipulation and functions from part1:

1. input a list of strings as a Universe.
2. takes another input a number of sets (that are subsets of the universe).
3. ask the user about the operations they want to perform

a) Union of two sets.

b) Intersection of two sets.

c) Complement of a set.

Part3: Applications for bits manipulation:

1. Find that single number by giving a non-empty array of integers numbers, every element appears twice except for one.

The solution with a linear runtime complexity and only constant extra space That takes an unsigned integer and returns the number of ’1’ bits it.

* **Used data structures:**

we used an array data structure:

part 1:

string universal[1000] ⮕ this array is for the universal set and it is of type string, we assumed random size at the begging as we don’t take the size of the array from the user.

int ArrSets[numberofsets] ⮕ this array is used to store the integer numbers which reflect the binary representation of each subset

ArrSets[i]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| universal[0] | subset[1] | subset[2] | subset[3] | subset[4] | … |

part 2:

we use arr[1000] ⮕ to store integer elements we assumed random size at the begging as we don’t take the size of the array from the user.

* **Algorithms used documented using flow charts.**

**Diagram

Description automatically generatedPart 1:**

(if part\_number!=”1” then this will go to part\_number==”3”)

**Diagram

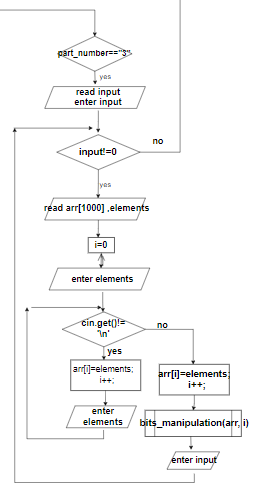
Description automatically generatedpart 2:**

(if part\_number!=”2” then this will go to part\_number==”3”)

**Diagram, schematic

Description automatically generated** (This will go to choose if he wants part 1 or 2 or 3 or exit)

**Part 3:**

****

(If the input == 0 then This will go to choose if he wants part 1 or 2 or 3 or exit)

* **Code Snippets.**

Part-1 functions:

Text

Description automatically generated with medium confidence

Part-2 functions:

Text

Description automatically generated with medium confidence

Part-3 functions:

Text

Description automatically generated

Subsets reading:

Graphical user interface, text, application

Description automatically generated

* **Sample runs and different test cases.**

**Part-1:**

**sample run (1):** get

2

0

Expected output: 0

Text

Description automatically generated

**sample run (2):** clear

5

2

Text

Description automatically generatedExpected output: 1

**sample run (3):** set

21

3

Expected output: 29

Text

Description automatically generated

**sample run (4):** update

10

2

Update value (Boolean): 1

Text

Description automatically generatedExpected output: 14

**sample run (5):** update

28

3

Update value (Boolean): 0

Expected output: 20

Text

Description automatically generated with medium confidence

**Part-2:**

**sample run (1):** 1 2 3 4 5

number of subsets:4

subset 1: 1 2

subset 2: 1

subset 3: 5 3

subset 4: 1 3 4

operation: union (number 1)

operation between :1 3

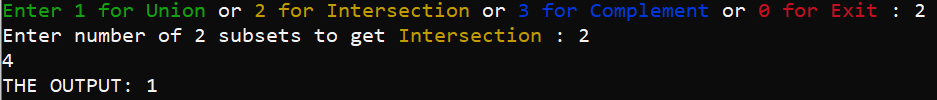
Text

Description automatically generatedExpected output: 1 2 3 5

operation: intersection (number 2)

operation between: 2 4

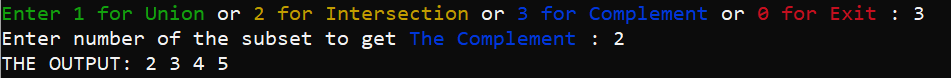
Expected output: 1



operation: complement (number 3)

operation on subset : 2

Expected output: 2 3 4 5



**sample run (2):** a b c d e f

number of subsets: 3

subset 1: a b d

subset 2: f

subset 3: e a f

operation: union (number 1)

operation between: 1 3

Expected output: a b d e f

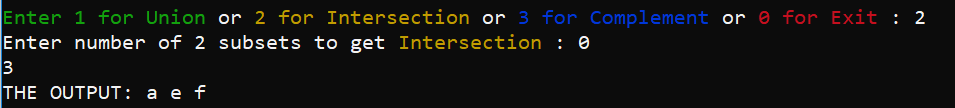
A screenshot of a computer

Description automatically generated with medium confidence

operation: intersection (number 2)

operation between: 0 3

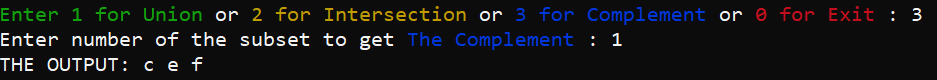
Expected output: e a f



operation: complement (number 3)

operation on subset : 1

Expected output: c e f



**sample run (3):** sara nancy nada

number of subsets: 3

subset 1: sara nada

subset 2: nancy sara

subset 3: NULL

operation: union (number 1)

operation between: 1 2

Expected output: sara nancy nada

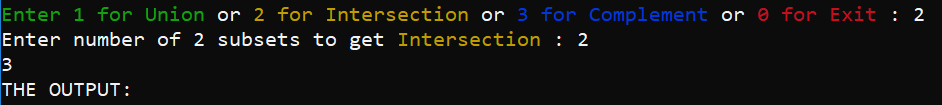
A screenshot of a computer

Description automatically generated with medium confidence

operation: intersection (number 2)

operation between: 2 3

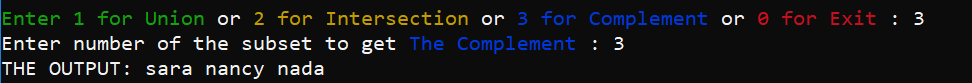
Expected output: (no expected output)



operation: complement (number 3)

operation on subset : 3

Expected output: sara nancy nada



**Part-3:**

**sample run (1):** 1 1 2 2 4

Text

Description automatically generated Expected output:4

**sample run (2):** 22 22 44 44 55

Expected output: 55

Text

Description automatically generated

* **Assumptions and details you find necessary to be clarified**

-we assumed that the user doesn’t enter the size of universal set or SUBSETS

-the user input is entered as follow:

1 2 3 4 5 -> this is a universal set each element is separated from another by whitespace and the user must press enter after the last element

-subsets are ordered according to the entered arrangement (1,2,3…)

-if we want to operate on universal set it is ordered 0