# Домашна работа 2

**Задача 1.**

Предварително са фиксирани 64 различни елемента. Всеки елемент е асоцииран с индекс от 0 до 63. Интересуваме се от множество, в което всеки елемент:

* Или присъства само веднъж
* Или отсъства

Наличието на елементи кодирайте в една променлива от тип **uint64\_t**.

* Ако елемент 5 присъства, то тогава 5ят бит е 1.
* Ако елемент 5 отсъства, то тогава 5ят бит е 0.

Възможните двоични представяния за подобно множество с 3 елемента са:

* 000 - всички елементи отсъстват
* 001 - само 3 присъства
* 010 - само 2 присъства
* 011 - 3 и 2 присъстват
* 100 - 1 присъства
* 101 - 1 и 3 присъстват
* 110 - 1 и 2 присъстват
* 111 - всички елементи присъстват

Реализирайте следните функции:

**#include <stdint.h>**

**#include <stdbool.h>**

**bool isEmpty(uint64\_t set);**

* Връща 1 ако множеството е празно. Иначе - връща 0.

**uint64\_t add(uint64\_t set, char index);**

* Добавя елемента с подадения индекс към множеството **set** и връща новото множество като резултат.

**uint64\_t remove(uint64\_t set, char index);**

* Премахва елемента с подадения индекс от множеството **set** и връща новото множество като резултат.

**bool isFull(uint64\_t set);**

* Връща 1 ако всички възможни елементи принадлежат на множеството. Иначе - връща 0.

**bool isSubset(uint64\_t setA, uint64\_t setB);**

* Връща 1 ако **setB** е подмножество на **setA.**

**uint64\_t union(uint64\_t setA, uint64\_t setB);**

* Връща обединението на **setA** и **setB.**

**uint64\_t intersect(uint64\_t, setA, uint64\_t setB);**

* Връща сечението на **setA** и **setB.**

**uint64\_t difference(uint64\_t setA, uint64\_t setB);**

* Връща разликата на **setA** и **setB.**

**char numElements(uint64\_t set);**

* Връща броя на елементите в **set**.

Използвайте побитови операции за да реализирате функциите, където това е възможно.