Soran University

Computer Science Department

OOP course

Homework #1

Deadline: 9/20/2023

1. Write a class Work whose objects represent working times (in whole minutes) and salary rates (in whole cents per minute). With this class, the following operations shall be possible:

```
Work w = new Work(25, 60);    // 25 cent/min, 60 min
w.add(65);                    // add 65 minutes working time
w.printSalary();              // prints salary "31,25" (25*125 Cents)
w.reset();                // reset working time to zero
bool okay = w.subtract(60);   // attempts to subtract 60 minutes
                               // returns false, if not sufficient
time
                               // available (time remains unchanged)
Work v = new Work(30); // 30 cent/min, 0 min
int r = w.compare(v); // 0 if salaries of w and v are equal,
                       // 1, if w’s salary is bigger, -1, else
```

2. Create a class IntegerSet for which each object can hold integers in the range 0 through 100. A set is represented internally as an array of ones and zeros. Array element a[i] is 1 if integer i is in the set. Array element a[j] is 0 if integer j is not in the set. The default constructor initializes a set to the so-called "empty set," i.e., a set whose array representation contains all zeros.

   (a) Write a unionOfSets member method that creates a third set that is the set-theoretic union of two existing sets (i.e., an element of the third set's array is set to 1 if that element is 1 in either or both of the existing sets, and an element of the third set's array is set to 0 if that element is 0 in each of the existing sets).
   (b) Write an intersectionOfSets member method which creates a third set which is the set-theoretic intersection of two existing sets (i.e., an element of the third set's array is set to 0 if that element is 0 in either or both of the existing sets, and an element of the third set's array is set to 1 if that element is 1 in each of the existing sets).
   (c) Write an insertElement member method that inserts a new integer k into a set (by setting a[k] to 1). Provide a deleteElement member method that deletes integer m (by setting a[m] to 0).
   (d) Write an isEqualTo member method that determines whether two sets are equal. If they are equal return true otherwise return false;

```
class Integerset
{
    int[] a = new int[101];
    public void unionOfSets(Integerset S1, Integerset S2);
    public void intersectionOfSets(Integerset S1, Integerset S2);
    public void insertElement(int k);
    public void deleteElement(int m);
    public bool isEqualTo(Integerset s);
    public Integerset()
    {
        for (int i = 0; i < 101; i++)
            a[i] = 0;
    }
}
```

```
        static void Main(string[] args)
        {
            Integerset myset1 = new Integerset();
            Integerset myset2 = new Integerset();
            Integerset myset3 = new Integerset();
            Integerset myset4 = new Integerset();
            myset1.insertElement(5);
            myset1.insertElement(51);

            myset2.insertElement(3);
            myset2.insertElement(5);
            myset2.insertElement(89);

            myset3.unionOfSets(myset1, myset2);  // myset3 includes 3,5,51,89;
            myset4.intersectionOfSets(myset1, myset2); //myset4 includes only 5;


            myset3.deleteElement(51); // element 51 is deleted from the set;

            myset3.isEqualTo(myset4); // return flase becuase myset4 is NOT equal to 
myset3.
        }
```