**Cairo University**

**Faculty of Computers & Information**

## Advanced Operating System (2014-2015)

## Assignment 1

It is needed to be familiar with the ***Deadlock*** problem and how to solve it. Therefore, it is required to apply the deadlock avoidance algorithms by applying the *Banker's algorithm*. You will write a Java code that implements the banker's algorithm: processes request and release resources and the banker will grant a request only if it leaves the system in a safe state. A request is denied if it leaves the system in an unsafe state.

The bank will employ the strategy outlined in the textbook whereby it will consider requests from n processes for m resources. The bank will keep track of the resources using the following data structures:

```
//the available amount of each resource   int [] available;

   //the maximum demand of each process   int [][] maximum;

int [][] allocation;//the amount currently allocated to each
                                                        process

  //the remaining needs of each process      int [][] need;
```

You should build a test program to test your code. The test program should take the initial number of the available resources at the bank and the maximum need for each process. The maximum need for the process should be stored in a text file. The file may contain a data as follow:

```
7,5,3
3,2,2
9,0,2
2,2,2
4,3,3
```

The file contains five rows for five processes. Each row represents the maximum needs for this process from the three available resources. The running statement could be as follow:

```
Java testBanker infile.txt 10 5 7
```

It means that the system has 10 instances from the first resource type and 5 instances from the second resource type and 7 from the third resource type.

**Output:**

The output screen must appear every action happened as well as the current state for the bank by showing the values for your data structures as well as the process requests and releasing for the resources. The bank decisions to either deny or approve the requests must be shown through the output screen.

Through the testing the user can type:

- " * " : show the current state information
- RL <process#> <r1> <r2> <r3>: it means that process number will release resources according to the values of the r1, r2, r3. E.g. "RL 2 2 3 1" means that process number 2 will release 2 instances from resource type 1, 3 instances from resource type 2, and 1 instances from resource type 3
- RQ <process#> <r1> <r2> <r3> : It means that process request resources
- Quit: it closes the application

**Submission**

This assignment will be submitted into groups of **two students**. Discussions will be held next week during lab times.

**Good Luck**