Dept. of Computer Science and Engineering
CS69011: Computing Lab 1

Assignment 6 — Maximum Marks: 30

The purpose of this assignment is to use semaphores to write a lift simulator program. The program simulates a number of lifts in a large building. The building is described by the following data structure:

```
typedef struct {
        int waitingtogoup;              /* the number of people waiting to go up */
        int waitingtogodown;            /* the number of people waiting to go down */
        semaphore up_arrow;             /* people going up wait on this */
        semaphore down_arrow;           /* people going down wait on this */
} Floor_info;
Floor_info floor[NFLOORS];
```

Each floor has two semaphores that people can wait upon, up_arrow and down_arrow.

LIFT:

A lift is described by the following data structure:

```
typedef struct {
        int no;                         /* which lift is it */
        int position;                   /* which floor it is on */
        int direction;                  /* going UP or DOWN */
        int peopleinlift;               /* number of people in the lift */
        int stops[NFLOORS];             /* for each stop how many people are waiting to get off */
        semaphore stopsem[NFLOORS];        /* people in the lift wait on one of these */
} Lift_info;
```

Each lift contains a semaphore for each floor that people inside can wait upon.

PERSON:

The algorithm for a person is:

```
while(1) {
        wait for a while
        pick a different floor to go to
        if going up
                press up arrow button and wait
        otherwise
                press down arrow button and wait
        get into lift
        press button for floor to go to and wait
        get out of lift
}
```

LIFT ALGORITHM:

The algorithm for a lift is:

```
while(1) {
        drop off all passengers waiting to get out at this floor
        if going up or empty
                if empty
                        direction=up
                        pick up all passengers on this floor waiting to go up
        if going down or empty
                if empty
                        direction=down
                        pick up all passengers on this floor waiting to go down
        move lift
        if on top or ground floors
                change direction
}
```

Your program must create a process for each lift and for a fixed number of people. Lifts go up and down, picking up people and taking them where they want to go. You need to add semaphore calls to communicate between processes and also for mutual exclusion.

A person starts on a certain floor and thinks of a random floor to travel to. The person then waits for a lift (going in the correct direction) by waiting on a semaphore ('up_arrow' or 'down_arrow') associated with its current floor. When the lift signals (performs a V operation) this semaphore, the person gets in and waits for the destination floor to be reached (by waiting on another semaphore). When the destination floor has been reached, the person gets out, waits a while and then starts another journey.

There are two semaphores on each floor, one for people going down and one for people going up (up_arrow and down_arrow). In every lift there is a semaphore for each floor (stopsem[]). The people in the lift wait on these before getting out.

Determine which are the shared variables. Use appropriate mutual exclusion when accessing the shared variables.

The time between successive trips of a person may be chosen using an exponential distribution. Some reference is provided below on how to do this.

You need to add delays to model a lift's movement from one floor to another. All lifts need not have the same speed.

An important component of this assignment is to show the output of the simulation in a way that one can see the movement of the lift(s), people waiting at different floors, and people getting in and getting out. An animation will be great, but other creative ways of displaying the dynamics will also be accepted.

Assuming that lifts have infinite capacity will be accepted, but if you consider lifts with finite capacity, it will be great (but more complicated).