CSC180 – Lab 3

Students: please show your TA your answers to 1 and 3 when you are done with those. As always, you’ll get credit for the lab if you make reasonable progress towards completing it: just do as much of the lab as you can.

1. Elevators are notoriously capricious. That’s partly because they’re really tricky to program well. Consider a building with two elevators and nFloors floors. (Define a global variable int nFloors to store the number of floors in the building.)

Write a function with the prototype

```c
int elevatorShouldStop(int floor, int upButton, int downButton, int e1From, int e1To, int e2From, int e2To);
```

The function returns 1 if elevator 1 should stop at floor `floor`, 0 if it shouldn’t, and -1 if the input is invalid. The other parameters are:

- `int e1From`: the floor from which elevator 1 is going
- `int e1To`: the floor to which elevator 1 is going
- `int e2From`: the floor from which elevator 2 is going
- `int e2To`: the floor to which elevator 2 is going
- `int downButton`: 1 if the DOWN button is pressed on floor `floor`, 0 otherwise
- `int upButton`: 1 if the UP button is pressed on floor `floor`, 0 otherwise

If elevator 1 (resp. 2) is not moving, then the floor on which the elevator is located = e1From = e1To (resp., e2From = e2To). An elevator should stop at a given floor if

- that floor is the elevator’s destination or
- The floor is “on the way,” and the appropriate elevator button has been pressed (i.e., up or down, depending on which way the elevator is going), and the other elevator is not currently in a non-moving state at that floor.

Devise a testing strategy and test this function.
2. Write a function that returns the number of days in a month, given the number of the month (1…12) and the year.

   Reminder:

   According to the **Gregorian calendar**, which is the civil calendar in use today, years evenly divisible by 4 are leap years, with the exception of centurial years that are not evenly divisible by 400. Therefore, the years 1700, 1800, 1900 and 2100 are not leap years, but 1600, 2000, and 2400 are leap years. (**Source: the US Naval Observatory website**) (Hint: use switch)

3a. Write a function with the signature `void nextDay(int y, int m, int d)`, which prints the date that follows the date y/m/d.

3b. Devise a testing strategy for the function `nextDay()`, and implement it.

4. Write a function that prints out, in order, all the dates between fY/fM/fD and tY/tM/tD.

5a. Implement a function which returns the number of days between fY/fM/fD and tY/tM/tD.

5b. Make the function from 5a as efficient as possible.