Simulation Lab no-6
Final Assignment -1 (Odd ID)

Problem 1:

Suppose you have a 5 * 5 array. Each cell of the array is either 0 (Dead) or 1 (Alive) . Now The value of a given cell at the next instant of time depends on the state of its neighbours at the previous time step. There are four rules:

1. If a cell is Alive and has fewer than two neighbours that are Alive, it dies on the next time step. (For Underpopulation)
2. If a cell is Alive and has either two or three neighbours that are alive, it remains Alive on the next time step.
3. If a cell is alive and has more than three neighbours that are Alive, it dies on the next time step. (For Overpopulation :3)
4. If a cell is Dead and has exactly three neighbours that are alive, it turns Alive on the Next Generation.

Example:

Time 0	Time -1	Time- 2	Time -3
0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0	0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0

So use the python programming language to solve this problem. Simulate your code up to 20-time steps and print the array at each time step. Use the above array at time = 0 for initialization.

Problem 2:

You have to simulate a chemical reaction involving three agents. At the beginning of the reaction two reagents A, B are present in the system with amounts of 50 units and 25 units respectively and C is present with amounts of 15 units. A and B react together and form C. Rate of the forward and backward reactions are .025 and .01 respectively. Now simulate the reaction using Python and report the time when the chemical reaction reaches equilibrium, the delta_t (time step) is 0.2 seconds and difference threshold is = 0.4. The equations for the rate of changes are given below:

$\frac{dA(t)}{dt}$=kb ∗ C(t) − kf ∗ A(t) ∗ B(t)

$\frac{dB(t)}{dt}$= kb ∗ C(t) −2* kf ∗ A(t) ∗ B(t)

$\frac{dC(t)}{dt}$ = 3* kf ∗ A(t) B(t) − 2* kb ∗ C(t)

where kf and kb are the rates of forward and backward reactions.

Output should be like this:

At time= 0.2 :
C1: 43.78
C2: 12.53
C3: 33.690000000000005
At time= 0.4 :
C1: 41.104563
C2: 7.111745999999999
C3: 41.783691000000005
At time= 0.6 :
C1: 39.726504324515005
C2: 4.272061267030019
C3: 46.001434408454976
At time= 0.8 :
C1: 38.96993689133561
C2: 2.666923531854318
C3: 48.36313957681007
At time= 1.0 :
C1: 38.54701396183732
C2: 1.7243513937041315
C3: 49.72863464445854
At time= 1.2 :
C1: 38.314128244885104
C2: 1.1591226905107805
C3: 50.526749064604104
At time= 1.4 :
C1: 38.19312786593538
C2: 0.8160684344821256
C3: 50.990803699582486
…..
…..
…..
…..
…..