Operating System

Name : TALEBUL ISLAM **Section :** GT

Registration No : 11904195 **Roll No :** 58

Email : talifanwarkhan@gmail.com

GitHub Link: https://github.com/talebulislam/Operating-System-Assignment

---

6. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use nonpreemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
P1	*0.0*	*8*
P2	*0.4*	*4*
P3	*1.0*	*1*

Solutions :

a. Ans :

FCFS Gantt Chart

```
Proc:  1               2      3
       |---------------|-------|----|
Time:  0               8      12   13
```

Average turnaround for these processes:

(8 + (12 - 0.4) + (13 - 1)) / 3 = 10.53

b. Ans :

SJF Gantt Chart

```
Proc:   1               3  2
        |---------------|--|--------|
Time:   0               8  9       13
```

Average turnaround for these processes:

(8 + (9 - 1) + (13 – 0.4)) / 3 = 9.53

c. Ans :

```
Proc:   x  3  2         1
        |--|--|---------|-----------------|
Time:   0  1  2         6                 14
```

CPU is left idle:

Average turnaround for these processes:

((2 - 1) + (6 – 0.4) + (14 - 0)) / 3 = 6.87

Q)21. A number of cats and mice inhabit a house. The cats and mice have worked out a deal where the mice can steal pieces of the cats' food, so long as the cats never see the mice actually doing so. If the cats see the mice, then the cats must eat the mice (or else lose face with all of their cat friends). There are NumBowls cat food dishes, NumCats cats, and NumMice mice. Your job is to synchronize the cats and mice so that the following requirements are satisfied:

No mouse should ever get eaten. You should assume that if a cat is eating at a food dish, any mouse attempting to eat from that dish or any other food dish will be seen and eaten. When cats aren't eating, they will not see mice eating. In other words, this requirement states that if a cat is eating from any bowl,then no mouse should be eating from any bowl. Only one mouse or one cat may eat from a given dish at any one time. Neither cats nor mice should starve. A cat or mouse that wants to eat should eventually be able to eat. For example, a synchronization solution that permanently prevents all mice from eating would be unacceptable. When we actually test your solution, each simulated cat and mouse will only eat a finite number of times; however, even if the simulation were allowed to run forever, neither cats nor mice should starve.

Solution →

In a rectangular field of size n by m squares there is a mouse and two cats. The mouse is the first to make a move, then each of the cats makes a move, then again its the mouse's turn, and so on. In each move both the mouse and the cats can move exactly one square vertically or horizontally. If the mouse is standing at the edge of the field then in its next move it can jump off the field and is saved from the cats. If in the next move one of the cats moves to the field with the mouse then there is no escape for the mouse

Function Description

Complete the *catAndMouse* function in the editor below. It should return one of the three strings as described.

catAndMouse has the following parameter(s):

- *x*: an integer, Cat A's position
- *y*: an integer, Cat B's position
- *z*: an integer, Mouse C's position

Input Format

The first line contains a single integer, q, denoting the number of queries.
Each of the q subsequent lines contains three space-separated integers describing the respective values of x (cat A's location),y (cat B's location), and z(mouse C's location).

Constraints

- 1<=q<=100
- 1<=x,y,z<=100

Output Format

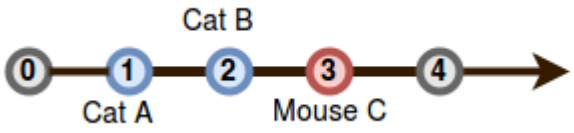
For each query, return Cat A if cat A catches the mouse first, Cat B if cat B catches the mouse first, or Mouse C if the mouse escapes.

Sample Output 0

```
C:\Users\Talib Khan\Desktop>java Cat_And_Mice
cat eating from dish 1
mice eating from dish 1
mice done eating
cat done eating


mice eating from dish 1
cat eating from dish 1
mice done eating

cat done eating

cat eating from dish 1
cat done eating

cat eating from dish 1
cat done eating

cat eating from dish 1
cat done eating

cat eating from dish 1
cat done eating


C:\Users\Talib Khan\Desktop>
```

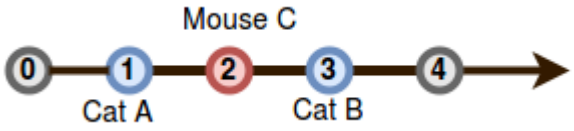
Explanation 0

Query 0: The positions of the cats and mouse are shown below:



Cat B will catch the mouse first, so we print Cat B on a new line.

Query 1: In this query, cats A and B reach mouse C at the exact same time:



Because the mouse escapes, we print Mouse C on a new line