True/False. Circle **only one** of T or F.

Use this table for questions 1-4 for deadlock avoidance questions based on the Banker’s algorithm. Treat each question independently, starting each time from the values shown in the table.

process max need allocated remaining need

A 10 4 6

B 9 2 7

C 8 3 5

Available units = 6

1. **T** / F Can safely grant a request from process A for 1 units.

2. **T** / F Can safely grant a request from process B for 1 unit.

3. T / **F** Can safely grant a request from process B for 2 units.

4. **T** / F Can safely grant a request from process C for 2 units.

Use this table for questions 5-8. Treat each question independently, starting each time from the values shown in the table.

process max need allocated remaining need

R1 R2 R3 R1 R2 R3 R1 R2 R3

A 2 3 4 1 1 1 1 2 3

B 4 3 2 2 2 2 2 1 0

C 6 5 4 3 3 3 3 2 1

Available units = 2 2 2

5. T / **F** Can safely grant a request from process A for 1 unit of R1

6. **T** / F Can safely grant a request from process B for 1 unit of R1

7. T / **F** Can safely grant a request from process C for 1 unit of R1

8. **T** / F Can safely grant a request from process A for 1 unit of R2

9. T / **F** An unsafe state will always lead to deadlock.

10. T / **F** If you prevent deadlock by making sure that one of the four conditions does not hold, you will also prevent starvation.