1. Start the program.
2. Initialise variables incoming, outgoing, buck\_size, n, and store to zero.
3. Prompt the user to enter the buck\_size (bucket size), outgoing rate, and n (number of inputs).
4. Read the values entered by the user for buck\_size, outgoing, and n.
5. Enter a loop that continues until n becomes zero.
6. Inside the loop, prompt the user to enter the incoming packet size.
7. Read the value entered by the user for incoming.
8. Check if incoming can be accommodated in the bucket buffer (incoming <= (buck\_size - store)).
9. If the condition is true, add incoming to the store (bucket buffer) and display the updated buffer size.
10. If the condition is false, calculate the number of packets to be dropped (incoming - (buck\_size - store)), display the dropped packets, and set the store to its maximum capacity (buck\_size).
11. Subtract the outgoing rate from the store to simulate the outgoing packets.
12. Display the number of packets left in the buffer after the outgoing packets are sent.
13. Decrement n by 1.
14. Repeat steps 5-13 until n becomes zero.
15. End the program.