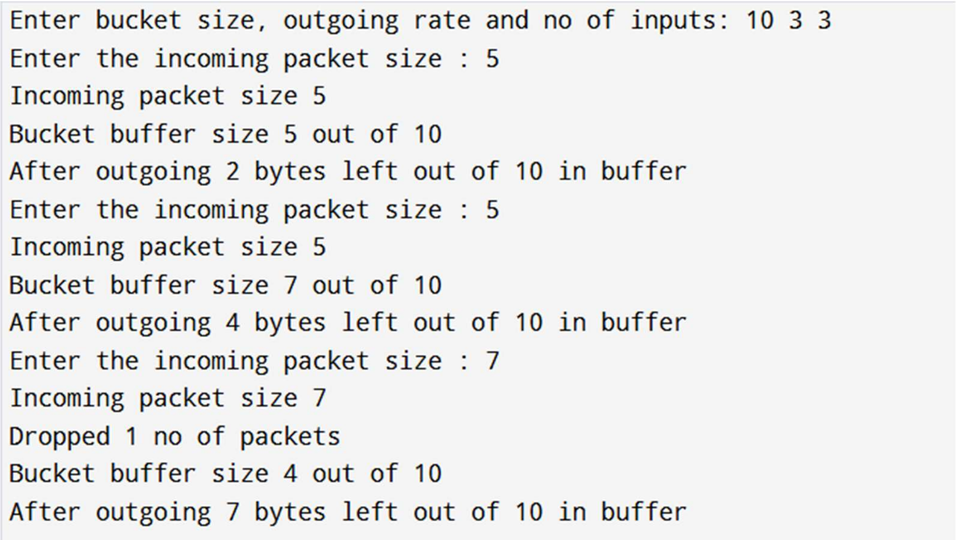
Program 14

Write a program for congestion control using Leaky Bucket algorithm

Code:

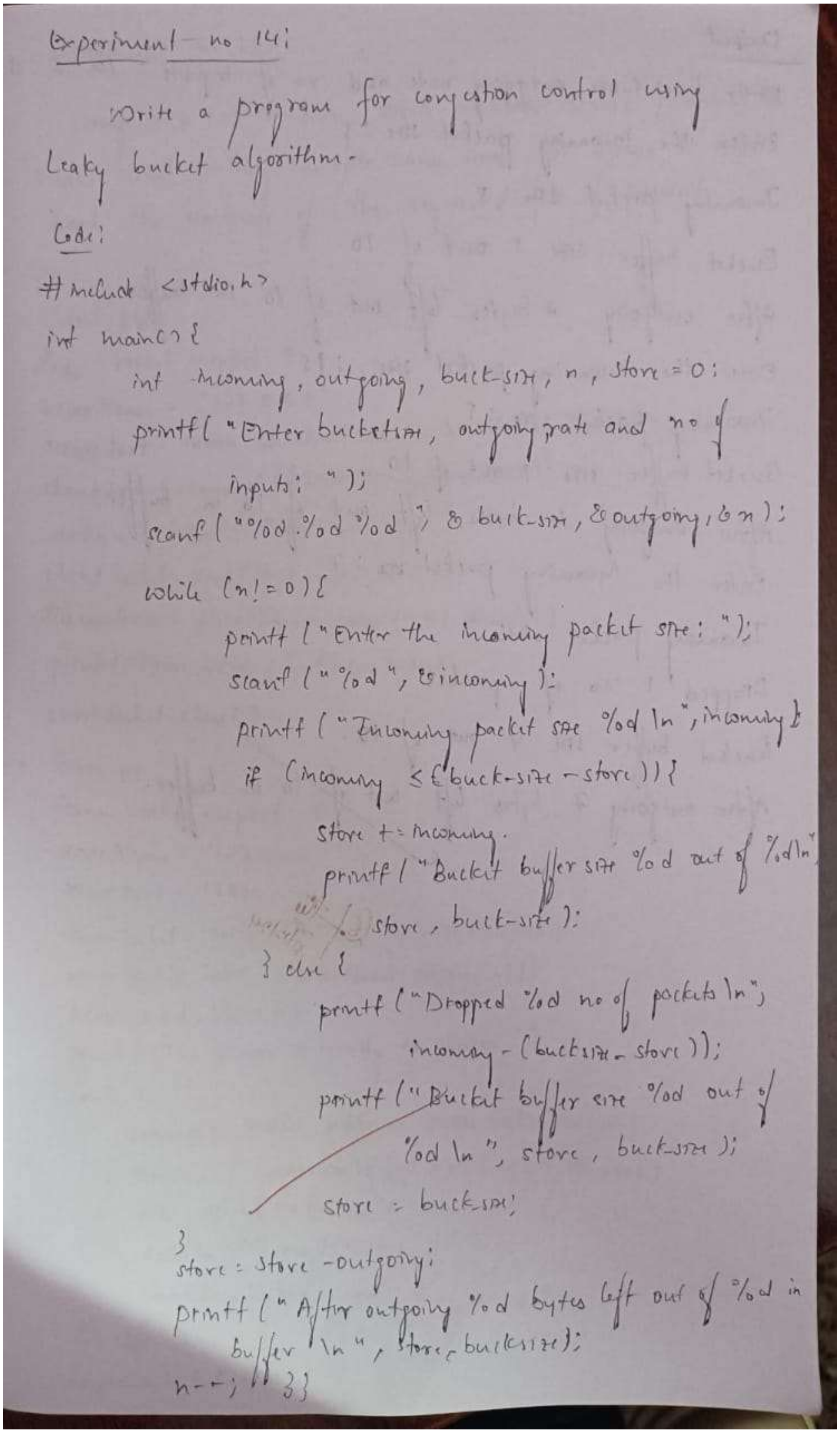
```
#include<stdio.h>

int main(){
    int incoming, outgoing, buck_size, n, store = 0;
    printf("Enter bucket size, outgoing rate and no of inputs: ");
    scanf("%d %d %d", &buck_size, &outgoing, &n);

    while (n != 0) {
        printf("Enter the incoming packet size : ");
        scanf("%d", &incoming);
        printf("Incoming packet size %d\n", incoming);
        if (incoming <= (buck_size - store)){
            store += incoming;
            printf("Bucket buffer size %d out of %d\n", store, buck_size);
        } else {
            printf("Dropped %d no of packets\n", incoming - (buck_size - store));
            printf("Bucket buffer size %d out of %d\n", store, buck_size);
            store = buck_size;
        }
        store = store - outgoing;
        printf("After outgoing %d bytes left out of %d in buffer\n", store, buck_size);
        n--;
    }
}
```

Output:



```
Enter bucket size, outgoing rate and no of inputs: 10 3 3
Enter the incoming packet size : 5
Incoming packet size 5
Bucket buffer size 5 out of 10
After outgoing 2 bytes left out of 10 in buffer
Enter the incoming packet size : 5
Incoming packet size 5
Bucket buffer size 7 out of 10
After outgoing 4 bytes left out of 10 in buffer
Enter the incoming packet size : 7
Incoming packet size 7
Dropped 1 no of packets
Bucket buffer size 4 out of 10
After outgoing 7 bytes left out of 10 in buffer
```

Figure 80: Output for Leaky Bucket algorithm



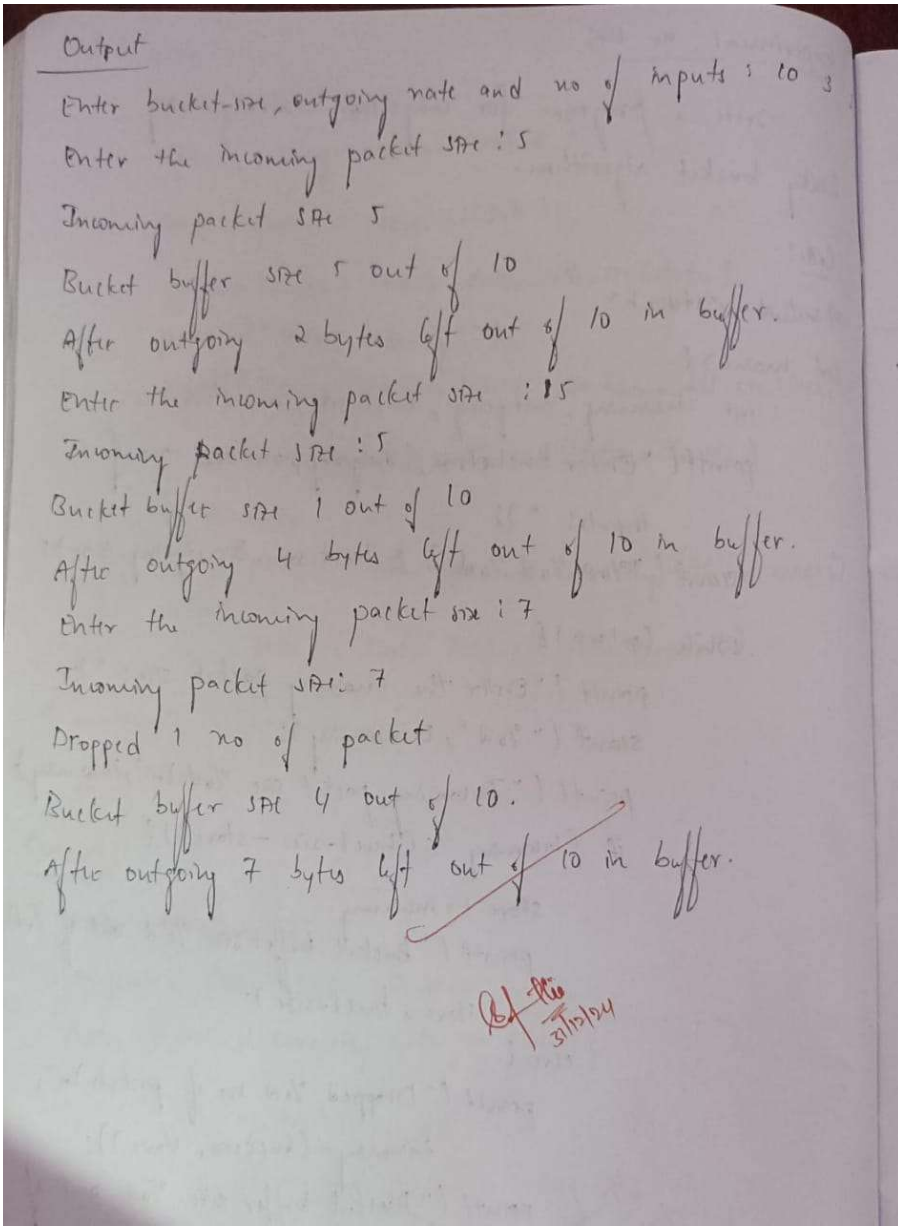
Experiment no 14:

Write a program for congestion control using Leaky bucket algorithm.

Code:

```
#include <stdio.h>
int main(){
    int incoming, outgoing, buck-size, n, store = 0;
    printf( "Enter bucketsize, outgoing rate and no of
            inputs: ");
    scanf ( "%d %d %d ", & buck-size, & outgoing, & n);

    while (n!=0){
        printf ("Enter the incoming packet size: ");
        scanf ("%d", &incoming):
        printf ("Incoming packet size %d \n", incoming);
        if (incoming <= (buck-size - store)){
            store += incoming.
            printf ("Bucket buffer size %d out of %d\n",
                store, buck-size);
        } else {
            printf ("Dropped %d no of packets \n",
                    incoming - (bucksize - store));
            printf ("Bucket buffer size %d out of
                    %d \n", store, buck-size);
            store = bucksize;
    }
    store = store - outgoing;
    printf (" After outgoing %d bytes left out of %d in
        buffer \n", store, bucksize);
    n--; }}
```

Figure 81: Observation Book 1



Output

```
Enter bucket-size, outgoing rate and no of inputs : 10 3
Enter the incoming packet size :5
Incoming packet size 5
Bucket buffer size 5 out of 10
After outgoing 2 bytes left out of 10 in buffer.
Enter the incoming packet size :5
Incoming packet size :5
Bucket buffer size 1 out of 10
After outgoing 4 bytes left out of 10 in buffer.
Enter the incoming packet size :7
Incoming packet size: 7
Dropped 1 no of packet
Bucket buffer size 4 out of 10.
After outgoing 7 bytes left out of 10 in buffer.
```

31/12/24

Figure 82: Observation Book 2