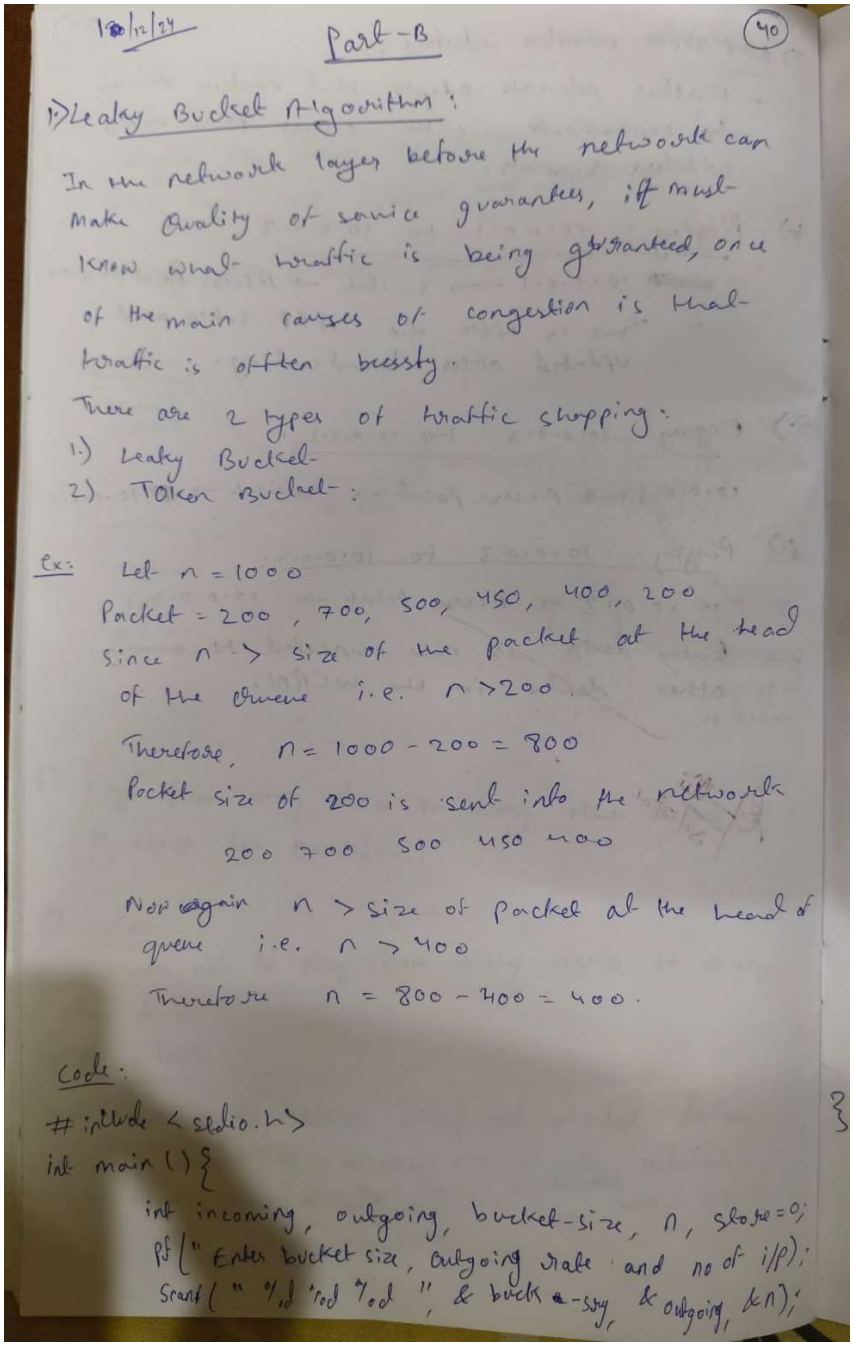
Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code :



Part-B (40)

1) Leaky Bucket Algorithm :

In the network layer before the network can make Quality of service guarantees, it must know what traffic is being guaranteed, one of the main causes of congestion is that traffic is often bursty.

There are 2 types of traffic shopping :

1.) Leaky Bucket-
2) Token Bucket :

Ex: Let n = 1000

Packet = 200, 700, 500, 450, 400, 200

Since n > size of the packet at the head of the queue i.e. n > 200

Therefore, n = 1000 - 200 = 800

Packet size of 200 is sent into the network

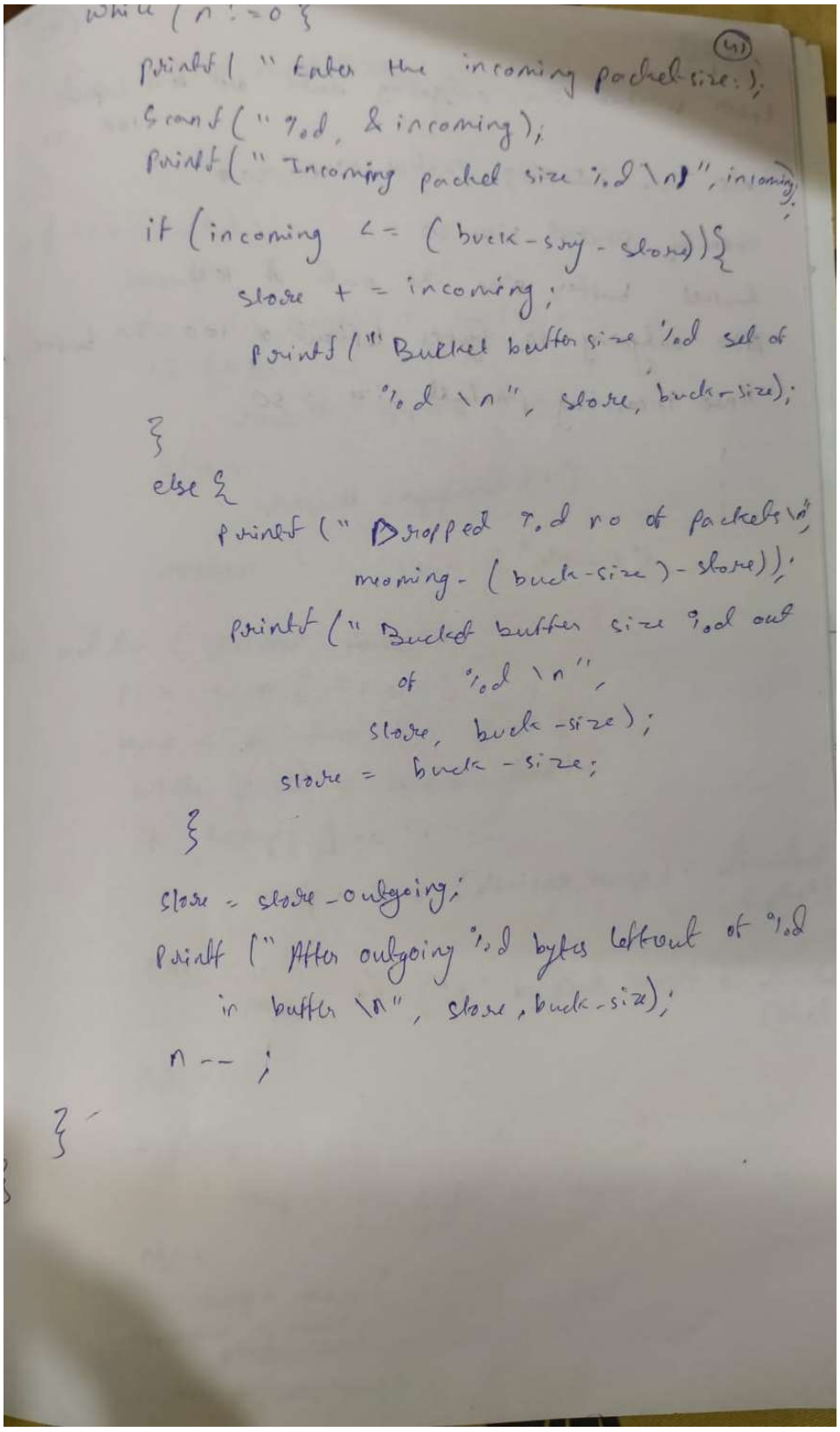
200 700 500 450 400

Now again n > size of packet at the head of queue i.e. n > 400

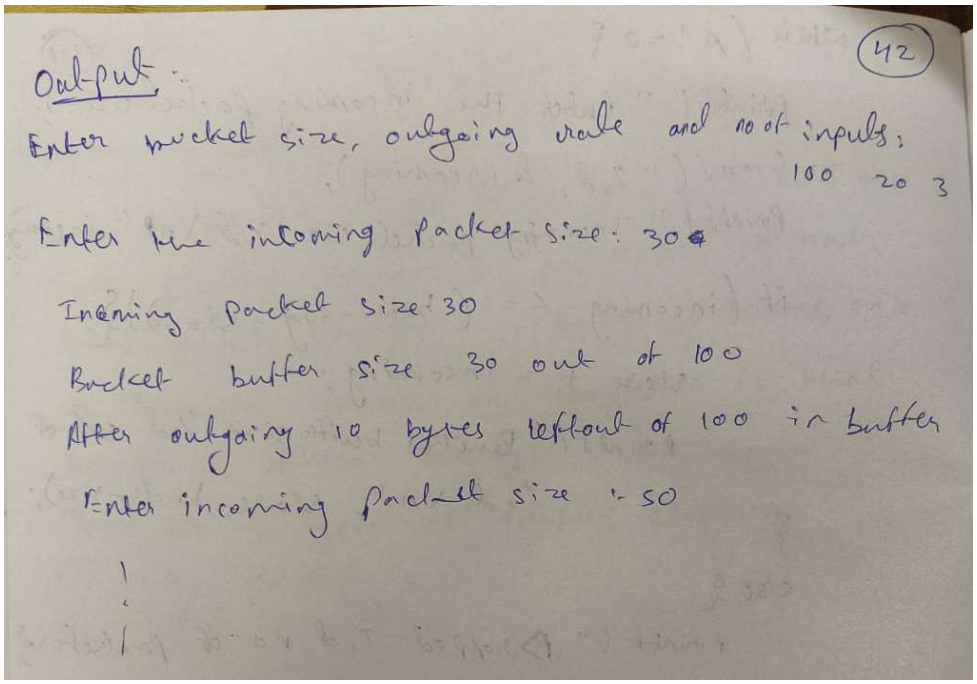
Therefore n = 800 - 400 = 400.

Code :

```
#include <stdio.h>
int main(){
    int incoming, outgoing, bucket_size, n, store=0;
    Pf(" Enter bucket size, outgoing rate and no of i/p);
    Scanf(" %d %d %d ", & bucket_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-stg - 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 leftout of %d
        in buffer\n", store, buck-size);
    n--;
}
```



Output :-

(42)

Enter bucket size, outgoing rate and no of inputs: 100 20 3

Enter the incoming packet size: 30

Incoming packet size 30

Bucket buffer size 30 out of 100

After outgoing 10 bytes leftout of 100 in buffer

Enter incoming packet size : 50

Output Clear

```
Generated packets: [80, 63, 57, 12, 69]
Enter bucket size: 60
Enter output rate: 30
Packet of size 80 bytes exceeds bucket capacity (60 bytes) - REJECTED
Packet of size 63 bytes exceeds bucket capacity (60 bytes) - REJECTED

Packet of size 57 bytes added to bucket
Bytes in bucket: 57
Transmitting 30 bytes
Bytes remaining in bucket: 27
Transmitting 27 bytes
Bytes remaining in bucket: 0

Packet of size 12 bytes added to bucket
Bytes in bucket: 12
Transmitting 12 bytes
Bytes remaining in bucket: 0
Packet of size 69 bytes exceeds bucket capacity (60 bytes) - REJECTED

=== Code Execution Successful ===
```