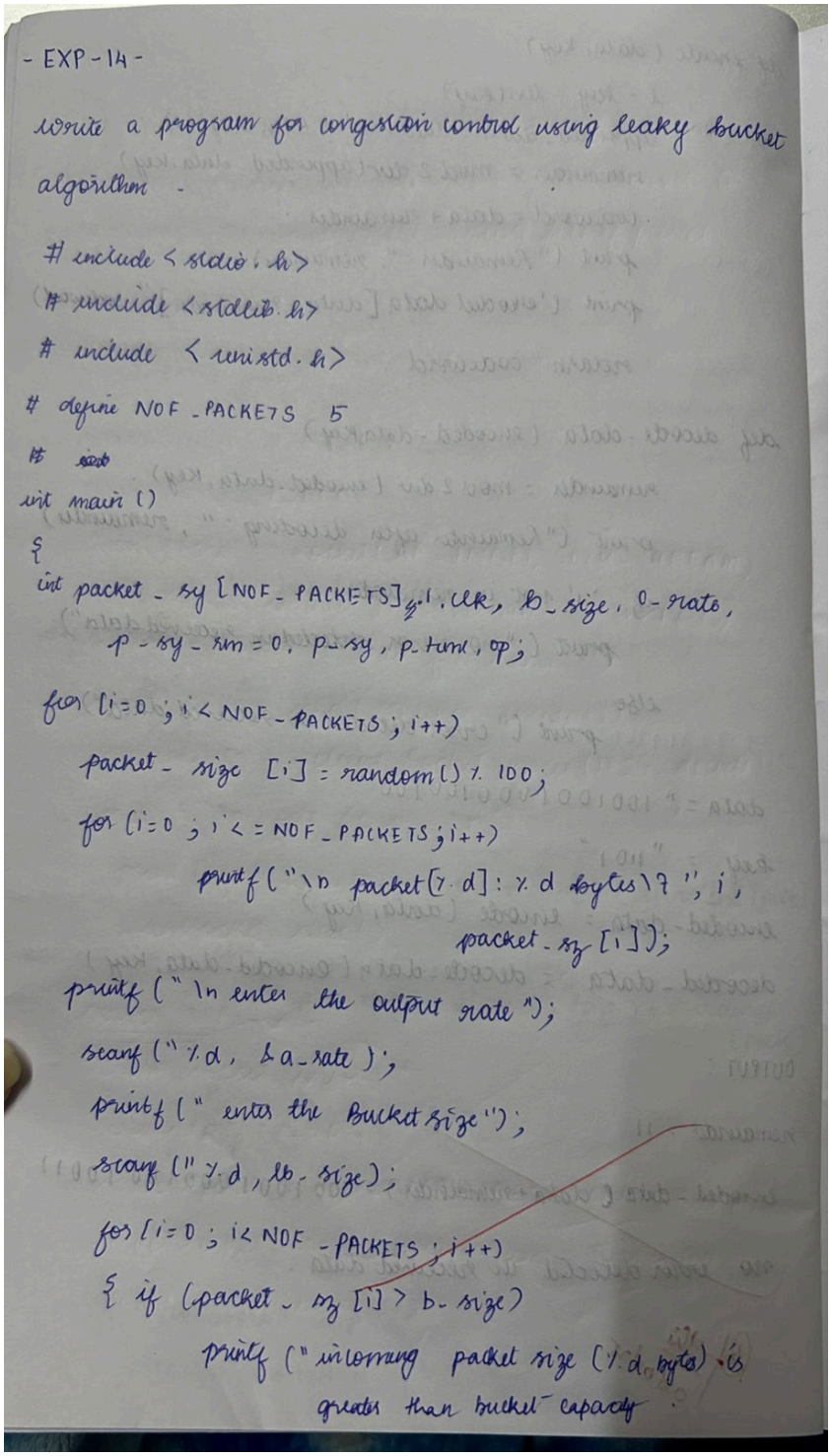
EXP-14

Write a program for congestion control using Leaky bucket algorithm.

Observation:



```
- EXP - 14 -

write a program for congestion control using leaky bucket
algorithm .

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define NOF_PACKETS 5
#
int main ()
{
int packet_sz [NOF_PACKETS], i, clk, b_size, o_rate,
    p_sz_rm = 0, p_sz, p_time, op;

for (i=0 ; i < NOF_PACKETS ; i++)
   packet_size [i] = random() % 100;
   for (i=0 ; i <= NOF_PACKETS ; i++)
          printf ("\n packet[%d]: %d bytes\t", i,
                                   packet_sz [i]);
  printf ("\n enter the output rate");
   scanf ("%d, &o_rate );
   printf (" enter the Bucket size");
   scanf ("%d, b_size);
   for (i=0 ; i< NOF_PACKETS ; i++)
   { if (packet_sz [i] > b_size)
          printf ("incoming packet size (%d bytes) is
                   greater than bucket capacity
```

bucket

(%d bytes) is greater
than bucket capacity).

```
else
    printf("bucket capacity exceeded packets rejected")

else {
    p_sz - rmt = packet_sz[i];
    printf("incoming packet size : %d", packet_sz[i]);
    printf("\n Bytes remaining to transmit ! %d",
                                    p_sz - rm);

    p_time = random() % 10;
    printf("time for transmission : %d unit", p_time);
    for(clk=10; clk <= p_time; clk+=10)
    while(p_sys_rm > 0)
        { sleep(1);
          if (p_size_rm)
            { if (p_sz_rm <= 10_rate).
                op = p_sz - rm, p_sz - rm = 0.
          else
                op = 0_rate, p_sz_rm = 0_rate
                printf( Packet of size %d transmitted op);

    }
    else
        printf("no packets to transmit!");
}
}
}
}
```

ftu
03/01/25

Code:

```
storage=0
noofqueries=int(input("Enter no of queries:"))
bucketsize=int(input("Enter bucket size:"))
inputpktsize=int(input("Enter input packet size:"))
outputpktsize=int(input("Enter output packet size:"))
for i in range(0,noofqueries):
    sizeleft=bucketsize-storage
    if inputpktsize<=sizeleft:
        storage+=inputpktsize
    else:
        print("Packet loss=", inputpktsize)
    print(f"Bucket size={storage}out of bucket size={bucketsize}")
    storage-=outputpktsize
```

Output:

```
Enter initial packets in the bucket: 0
Enter total no. of times bucket content is checked: 4
Enter total no. of packets that can be accommodated in the bucket: 10
Enter no. of packets that enters the bucket at a time: 4
Enter no. of packets that exits the bucket at a time: 1
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```