```
write a program for congestion control resing keeky
bucket algorithm
 #include stolio. n>
  void main()
     int incoming, out going, buck size, n stoke o:
    printl ("Enter bucket size, outgoing note and no of inputs")
     scary ("7.1 7.1 7.1", 2 buckesize ,2 outgoing, 4n); c ;
     while (n! = 0) sns (no - (n+3) + (n) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3) + (3)
        print ("Inter the incoming packet size:"):
         scary ("y.d", Lincorning);
         points (" Incoming packet size x.d", incoming)
           4 (incoming <= (buck-siz -stote)) & some of the
                             ston t= incoming ? " " microge of
                             print ("Bucket buffer size i'd out of i'd in import
                                         buck-size);
                       I else & peremarke nellier
                     printy ("Dropped id no of packets incomes - (bud six
                     store));
                   print ("Bucket buffer size rd out of rd" stole,"
                        buck_size);
                         Store-buck size } been in many morning
                Share - share outgoing;
                 print 1. After outgoins 1.d packets left out of 1.d in
                 buffer", stole, buck_size);
                    n -- ;
 Output:
 Enter bucket size, outing nate & no of inputs . 20 10 2
 Enter the incoming packet size: 30
Incoming packet size: 30
  Doopped 10 no of packets
 Buffer size 0 out of 20
```

Appen outgoing to packets left out of 20 in buffer knowning packet size: 10

Incoming packet size: 10

Buffer size 20 out of aw 20

After outgoing to packets out of 20 in buffer

After outgoing to packets out of 20 in buffer

## **Output:**

Enter bucket size, outgoing rate and no of inputs: 10 10 2 Enter the incoming packet size : 30

Incoming packet size 30

Dropped 20 no of packets

Bucket buffer size 0 out of 10

After outgoing 0 packets left out of 10 in buffer

Enter the incoming packet size : 10

Incoming packet size 10

Bucket buffer size 10 out of 10

After outgoing 0 packets left out of 10 in buffer