Program-4 Leaky-Bucket Algorithm for Congestion control

```
import os
clear = lambda: os.system('clear')

class client:
    def _init_(self, rate = int, data = []):
        self.rate = rate
        self.data = data

    def _str_(self):
        return str([str(self.rate), str(self.data)])


class Buffer:
    def _init_(self, buffer_size = int, buffer = []):
        self.buffer_size = buffer_size
        self.buffer = buffer

    def checkstate(self):
        if len(self.buffer)==0:
            return True

    def _str_(self):
        return str([str(self.buffer_size),
                    str(self.buffer)]);
```

```
basestate = True
Sec = 1
buffer = Buffer(int(input("Enter buffer size")))
client = Client( int (input("Enter client acceptance
                                   rate in bps")))

data_to_send = Str

while basestate:

  data_to_send = input("Enter a String send by the
                                   Server")
  count = 0
  if buffer.checkstate():
        for i in range(0, len(data_to_send)):
           if  i < client.rate:
                cliend.data.append(data_to_send[i])
           else:
               if count < buffer.buffer_size:
                   buffer.buffer.append(data_to_send[i])
                   count = len(buffer.buffer)
               else:
                   print("Data loss" + data_to_send[i])

  else:
      j=0
      for i in range(0, len(data_to_send)+ len(
                                   buffer.buffer)):
        if  i < client.rate:
            if len( buffer.buffer):
                client.data.append( buffer.buffer[0])
                del buffer.buffer[0]
        else:
            client.data.append(data_to_send[j])
                j += 1
```

```
else:
    if len(buffer.buffer) <= buffer.buffer_siz
        if j < len(data_to_send):
            buffer.buffer.append(data_to_send[j])
            j+ =1
        else:
            if j < len(data_to_send):
                print("Data loss"+ data_to_send[j]
                j+ =1;

print(buffer)
print(client)
```