

No:

Date:

CN LAB-8 LEAKY BUCKET ALGORITHM

```
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
```

```
baseState = True
```

```
Sec = 1
```

```
buffer = Buffer(int(input("Enter buffer size: ")))
```

```
client = Client(int(input("Enter rate: ")))
```

```
data_to_send = str
```

No:

Date:

```
while basestate:
    data_to_send = input("Enter string: ")
    count = 0
    if buffer.checkstate():
        for i in range(0, len(data_to_send)):
            if i < client_rate:
                client.data.append(data_to_send[i])
                use
                if count < buffer.buffer_size:
                    buffer.buffer.append(data_to_send[i])
                else:
                    if count < buffer.buffer_size:
                        data_to_send[i]
                    else:
                        j = 0
            for i in range(0, len(data_to_send) + len(buffer.buffer)):
                if i < client_rate:
                    client.data.append(buffer.buffer[0])
                else:
                    client.data.append(data_to_send[i])
                    j += 1
                    else: if len(buffer.buffer) <=
                        buffer.buffer.append(data_to_send[i])
                        j += 1
                    else: if j < len(data_to_send):
                        Print("Data loss: " + data_to_send[j])
                        j += 1;
                        Print(buffer)
                        print(client)
```

