Leaky Bucket Algorithm

P. SAI DEEKSHITH
1BM18CS148

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int BucketSize;
    cout << "Enter the max bucket size : ";
    cin >> BucketSize;
    int OutputRate;
    cout << "Enter the O/P rate : ";
    cin >> OutputRate;
    cout << "Enter the Packets : ";
    int numberOfPackets;
    cin >> numberOfPackets;
    for(int i=0; i< numberOfPackets; i++)
    {
        int packetSize;
        cout << "Enter the Packet Size : ";
        cin >> packetSize;
        leakyBucket(OutputRate, BucketSize,
                    packetSize);
    }
}
```

```
Void leakyBucket (int outputRate, int BucketSize,
                  int PacketSize)
{
if (PacketSize <= OutputRate)
{
cout << "Bucket Output is sufficient" <<
cout << "Last " << PacketSize << " was
        sent " << endl;
}
else if (PacketSize <= BucketSize)
{
cout << "Bucket Output is sufficient";
cout << "Last " << OutputRate << "
        bytes was sent " << end;
int rem_size = PacketSize - Output
                            Rate;
while (rem_size > 0)
{
    if (rem_size >= OutputRate)
    cout << "Last " << OutputRate
         << "Bytes was sent"
    else
    Cout << "Last " << rem_size
         << " was sent";
    rem_size = rem_size -
               outputRate;
}
else cout << " BUCKET OVERFLOW! ";
}
```