Samir Kumar
1BM18CS091

Page No.
Date : / /

leaky bucket algorithm

```
#include <bits/stdc++.h>
#include <unistd.h>
using namespace std;
#define bucketsize 500

void bucketInput(int a, int b)
{
    if (a > bucketsize)
        cout << "\n\t\t Bucket overflow";
    else {
        sleep(5);
        while (a>b) {
            cout<<"\n\t\t" <<b<<" bytes
                                outputted.";
            a -= b;
            sleep(5);
        }
    if (a>0)
        cout<<"\n\t\t Last " <<a<<" bytes sent\t";
    cout <<"\n\t\t Bucket output successful";
    }
}
int main()
{
```

```
{ int op, pkt size;
cout << "Enter output rate : ";
cin >> op;
for (int i=1; i<=5 ;i++)
{ sleep(rand()%10);
  pktsize = rand()%700;
  cout << "\n Packet no" << i << "\t Packet Size =
                          " << pktsize;

  bucketInput(pktsize, op);
}
cout<<endl;
return 0;
}
```