Subject :	Software :
	Hardware :
Branch : / Semester :	Page No. / Prog No. 7

PROBLEM STATEMENT

Write a program to congestion control using leaky bucket algorithm.

ALGORITHM & CODE :

The leaky Bucket is used for traffic shaping and congestion control in networking.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define BUCKET_SIZE 5
#define LEAK_RATE 1

typedef struct {
    int currentWater;
    int leakRate;
    int bucketSize;
} leackyBucket;

void initBucket (leackyBucket * bucket, int size, int leakRate)
{
    bucket -> currentWater = 0;
    bucket -> bucketSize = size;
    bucket -> leakRate = leakRate;
}

void addData (leackyBucket * bucket, int data){
    if (bucket -> currentWater + data > bucket -> bucketSize)
{   Printf ("Bucket overflow ! Discarding excess data . \n");
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty
Date :

Signature of Student
Date :

Subject :	Software :
	Hardware :
Branch : / Semester :	Page No. / Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
        bucket -> currentWater += data;
        Printf ("Added %d units of data to the bucket. Current
Water level : %d \n", data, bucket -> currentWater);
    }
}

Void leakData (leakyBucket *bucket){
    if (bucket -> current water > 0){
        bucket -> currentWater -= bucket ->leakRate;
        if (bucket ->currentWater < 0){
            bucket -> currentWater = 0;
        }
        Printf ("Leaked %d units of data. Current
Water level : %d \n", bucket -> leakRate , bucket ->
currentWater);
    }else{
        printf ("No data to leak. \n");
    }
}
int main(){
    leackyBucket bucket;
    initBucket (& bucket ,BUCKET_SIZE , LEAK_RATE);

    int incomingData;
    while (1){
        Printf ("\n Enter incoming data (or -1 to exit): ");
        Scanf("%d", & incomingData);
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :	Signature of Faculty Date :	Signature of Student Date :

Subject :	Software :
	Hardware :
Branch : / Semester :	Page No. / Prog No.

LEM STATEMENT

RITHM & CODE :

```
        if (incomingData == -1) {
            break;
        }
        addData (&bucket, incomingData);
        leakData (&bucket);
        sleep(1);
    }
    return 0;
}
```

O/P

Enter incoming data (or -1 to exit): 4
Added 4 units of data to the bucket:
Current Water level: 4
Leaked 1 unit of data. Current water level: 3

Enter incoming data (or -1 to exit): 3
Bucket overflow! Discarding excess data.
Leaved 4 units of data. current water level: 4

Enter incoming data (or -1 to exit): -1

PUT GIVEN

TPUT OBTAINED

EMARKS

RADE :

Signature of Faculty
Date :

Signature of Student
Date :