

### Aim-14

14. Write a program for congestion control using Leaky bucket algorithm.

Code:

18/08/23

Aim-14

2. Write a program for congestion control using leaky bucket algorithm.

```
import java.util.Scanner;

public class LeakyBucketAlgorithm {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        int bucketSize, outgoingRate, totalTime;

        System.out.print("Enter the bucket size (in packets): ");

        bucketSize = scanner.nextInt();

        System.out.print("Enter the outgoing rate (Packets per second): ");

        outgoingRate = scanner.nextInt();

        System.out.print("Enter the total time (in seconds): ");

        totalTime = scanner.nextInt();

        int currentBucketSize = 0;

        int time = 0;
```

```
while (time < totalTime) {
```

```
    System.out.print("Enter packet size  
    (in packets): ");
```

```
    int packetSize = scanner.nextInt();
```

```
    if (packetSize <= bucketSize) {
```

```
        if (packetSize <= (bucketSize - currentBucketSize))  
        {
```

```
            currentBucketSize += packetSize;
```

```
            System.out.print("packet of size " + packetSize +  
                             " transmitted.");
```

```
            int remainingEmptySize = bucketSize - current  
                                     BucketSize;
```

```
            System.out.println("Bucket size: " + bucketSize +  
                               " packets");
```

```
            System.out.println("Remaining Empty size: " +  
                               remainingEmptySize + " packets");
```

```
        } else {
```

```
            System.out.println("packet discarded: Insufficient  
                                space in the bucket.");
```

```
        }
```

```

} else
    system.out.println("packet discarded: Exceeds
                        the bucket size.");
    time++;
}
Scanner.close();

```

Output 1:

Enter the bucket size (in packets): 5  
 Enter the outgoing rate (Packets per second): 2  
 Enter the total time (in seconds): 5  
 Enter the packet size (in packets): 3  
 packet of size 3 transmitted. Bucket size: 5  
 packets

Remaining Empty size: 2 packets  
 Enter packet size (in packets): 10  
 packet discarded: Exceeds the bucket size.  
 Enter packet size (in packets): 3  
 packet discarded: Insufficient space in the bucket



Enter packet size (in packets): 2

Packets of size 2 transmitted.

Bucket size: 5 packets

Remaining Empty size = 0 packets.

Output 2:

Enter the bucket size (in packets): 10

Enter the outgoing rate (packets per second): 2

Enter the total time (in seconds): 5

Enter packet size (in packets): 3

Packets of size 3 transmitted

Bucket size: 10 packets

Remaining Empty size: 7 packets

Enter packet size (in packets): 5

Packets of size 5 transmitted

Bucket size: 10 packets

Remaining Empty size: 2 packets

Enter packet size (in packets): 4

packet discarded: Insufficient space in the bucket

Enter packet size (in packets): 12

packet discarded: Exceeds the bucket size

Enter packet size (in packets): 2

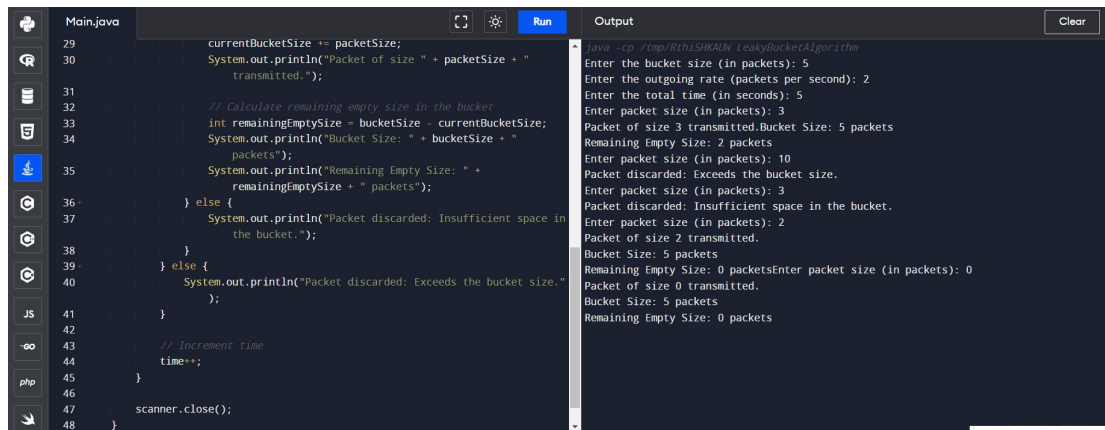
packet of size 2 transmitted.

Bucket size = 10 packets

Remaining Empty size: 0 packets.

NP  
11/9/2023

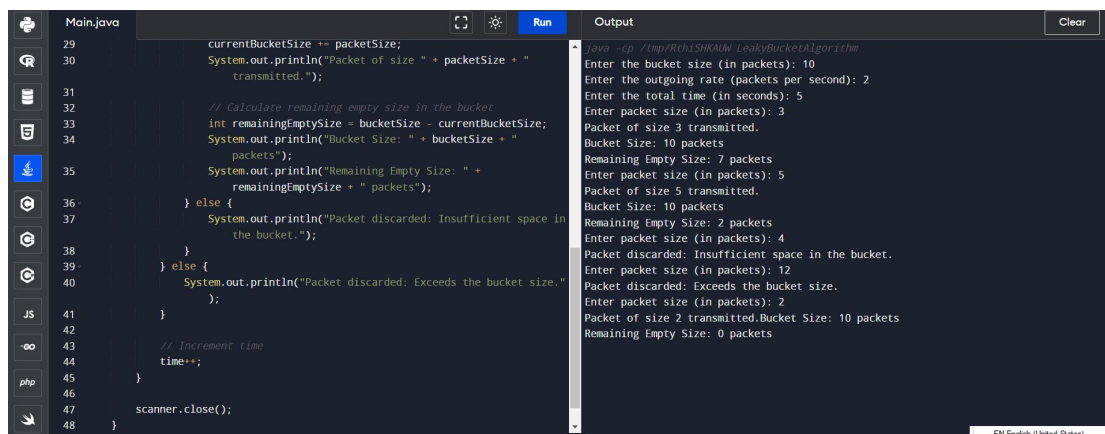
Output:



The screenshot shows an IDE with a file named 'Main.java' and an 'Output' window. The code implements a bucket algorithm simulation. The output window shows the following sequence of events:

```
java -cp ./tmp/Rth1SHKALW LeakyBucketAlgorithm
Enter the bucket size (in packets): 5
Enter the outgoing rate (packets per second): 2
Enter the total time (in seconds): 5
Enter packet size (in packets): 3
Packet of size 3 transmitted.Bucket Size: 5 packets
Remaining Empty Size: 2 packets
Enter packet size (in packets): 10
Packet discarded: Exceeds the bucket size.
Enter packet size (in packets): 3
Packet discarded: Insufficient space in the bucket.
Enter packet size (in packets): 2
Packet of size 2 transmitted.
Bucket Size: 5 packets
Remaining Empty Size: 0 packetsEnter packet size (in packets): 0
Packet of size 0 transmitted.
Bucket Size: 5 packets
Remaining Empty Size: 0 packets
```

Fig1: Output 1



The screenshot shows the same IDE with 'Main.java' and 'Output' window. The code is identical to the first screenshot. The output window shows a different sequence of events due to different input values:

```
java -cp ./tmp/Rth1SHKALW LeakyBucketAlgorithm
Enter the bucket size (in packets): 10
Enter the outgoing rate (packets per second): 2
Enter the total time (in seconds): 5
Enter packet size (in packets): 3
Packet of size 3 transmitted.
Bucket Size: 10 packets
Remaining Empty Size: 7 packets
Enter packet size (in packets): 5
Packet of size 5 transmitted.
Bucket Size: 10 packets
Remaining Empty Size: 2 packets
Enter packet size (in packets): 4
Packet discarded: Insufficient space in the bucket.
Enter packet size (in packets): 12
Packet discarded: Exceeds the bucket size.
Enter packet size (in packets): 2
Packet of size 2 transmitted.Bucket Size: 10 packets
Remaining Empty Size: 0 packets
```

Fig 2: Output 2