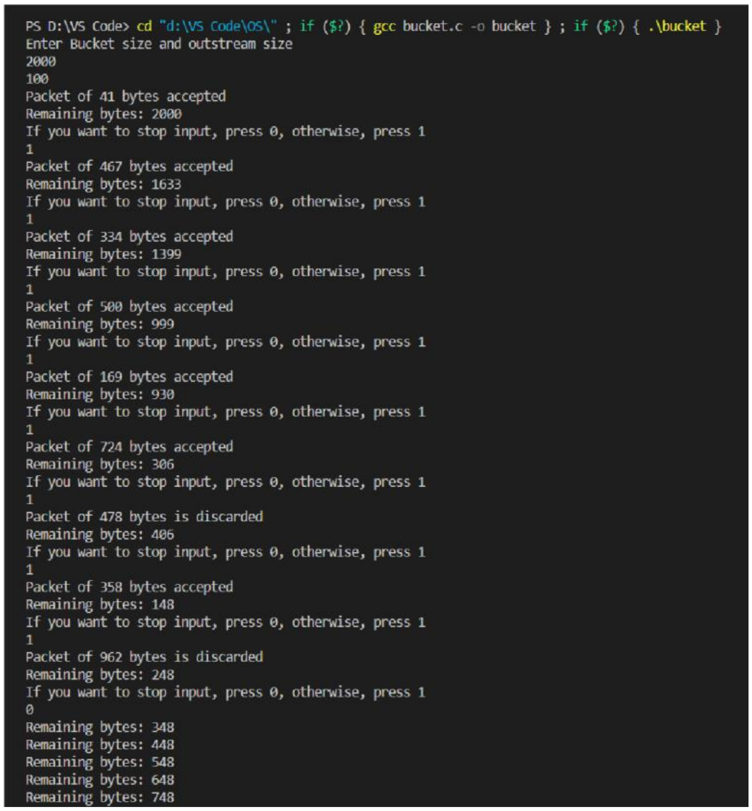
WEEK14

Write a program for congestion control using Leaky bucket algorithm.

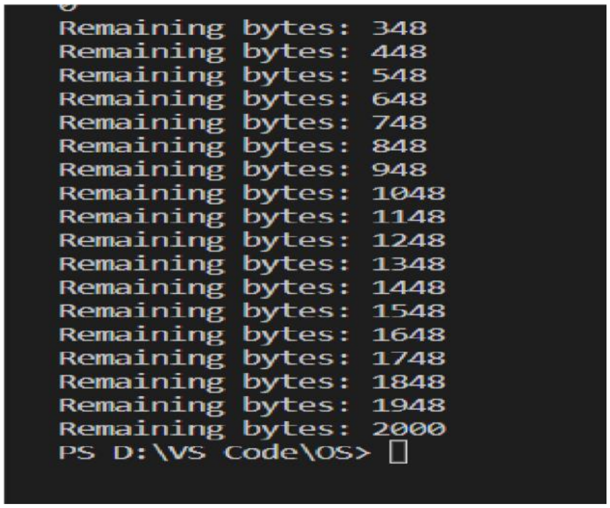
CODE:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function int
main()
{
    int buckets, outlets, k = 1, num, remaining;
    printf("Enter Bucket size and outstream size\n");
    scanf("%d %d", &buckets, &outlets); remaining =
    buckets; while (k)
    {
        num = rand() % 1000; // Generate a random number between 0 and
999 if (num < remaining)
        {
            remaining = remaining - num; printf("Packet of %d bytes
            accepted\n", num); // Added missing
variable
        }
        else
        {
            printf("Packet of %d bytes is discarded\n", num);
        }
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets; printf("Remaining
        bytes: %d \n", remaining); printf("If you want to
        stop input, press 0, otherwise, press 1\n");
        scanf("%d", &k);
    }
    while (remaining < buckets) // Fixed the condition
    {
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
    return 0; // Added a return statement to indicate successful completion
}
```

OUTPUT:

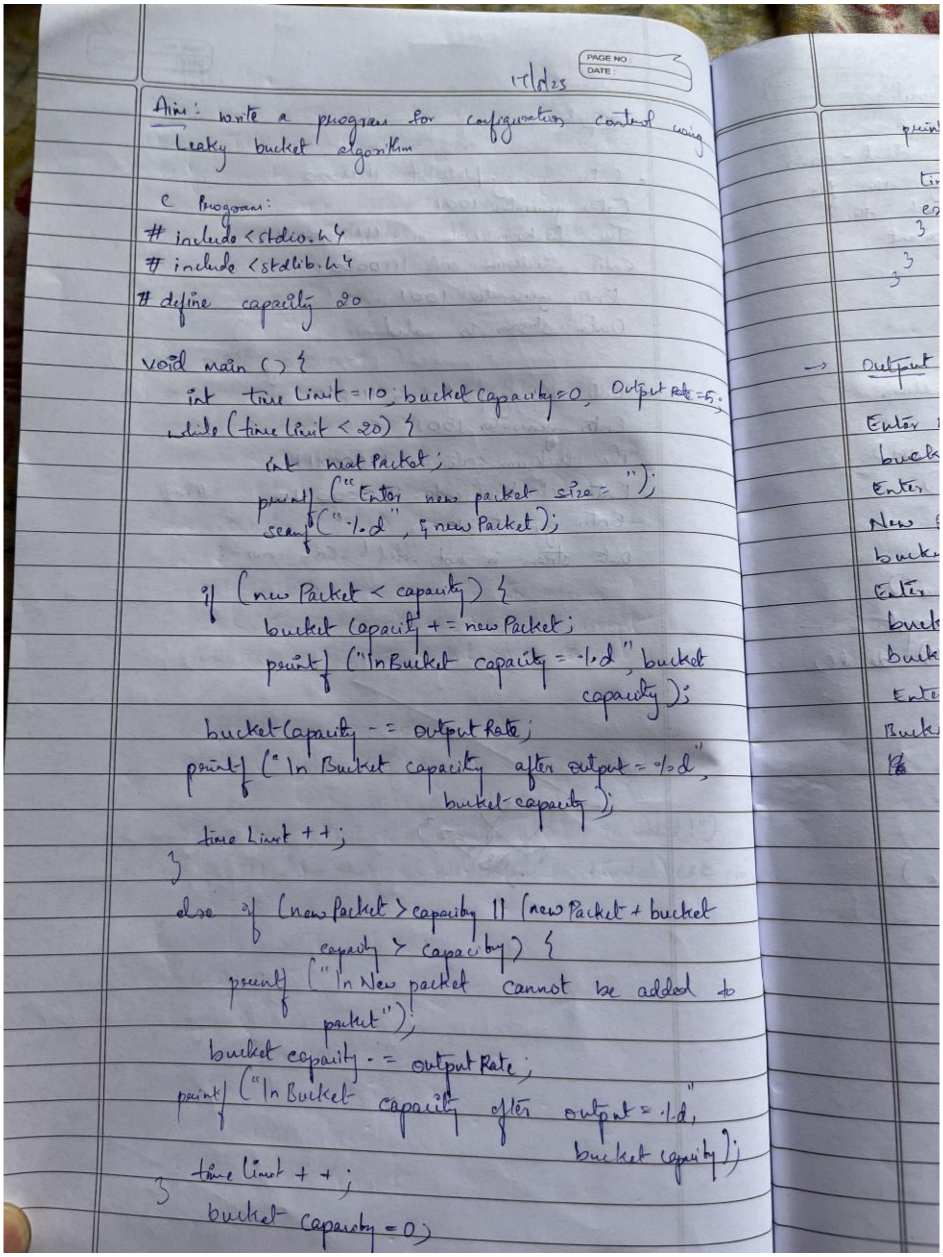


```
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
```



```
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS> 
```

OBSERVATION:



PAGE NO :
DATE :

17/8/23

Aim: write a program for configuration control using Leaky bucket algorithm

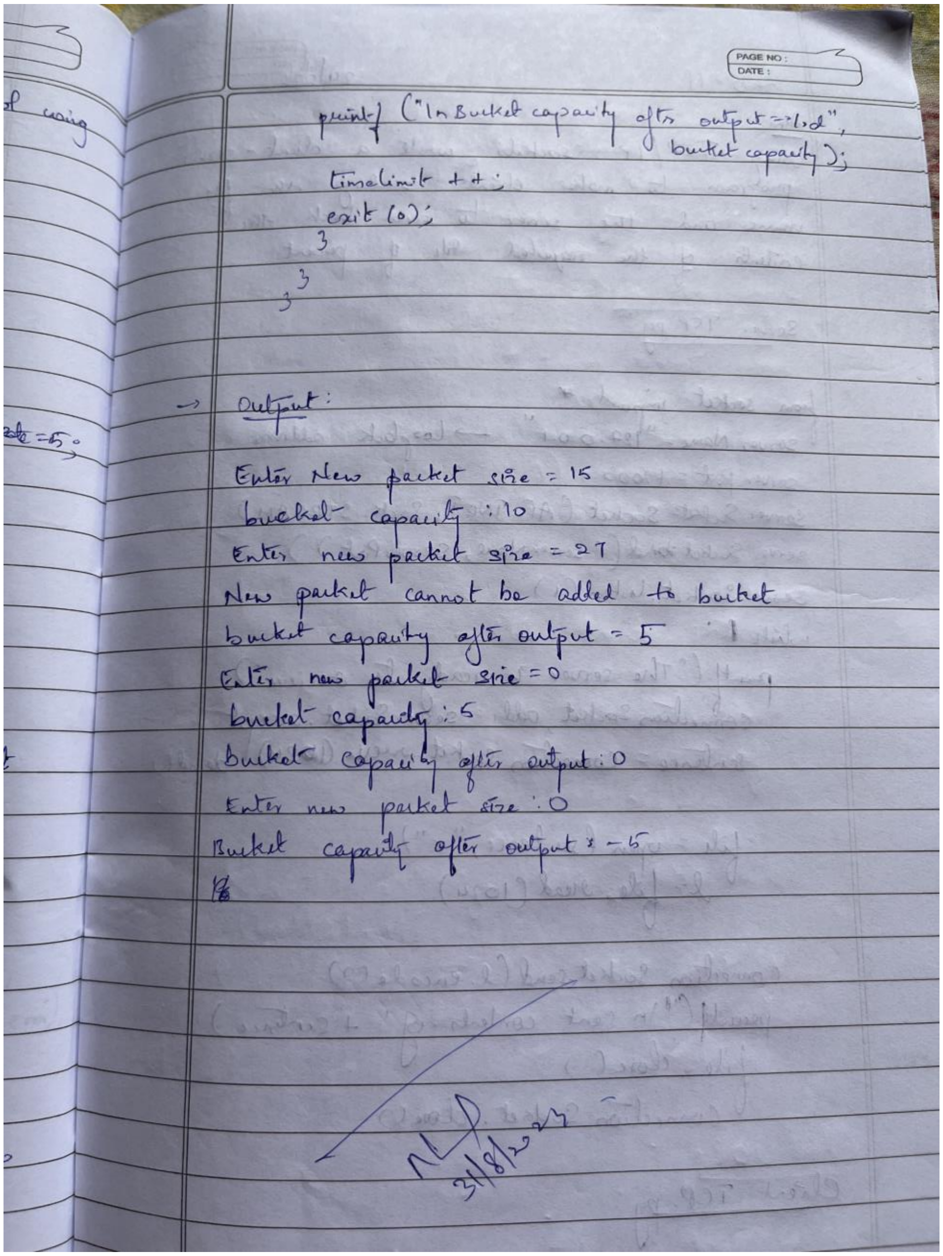
C Program:

```
# include <stdio.h>
# include <stdlib.h>
# define capacity 20

void main () {
    int time Limit = 10; bucket Capacity = 0, Output Rate = 5;
    while (time limit < 20) {
            int next Packet;
            printf ("Enter new packet size = ");
            scanf ("%d", & new Packet);

        if (new Packet < capacity) {
            bucket Capacity += new Packet;
            printf ("\nBucket capacity = %d", bucket
                                                capacity);
        bucket Capacity -= output Rate;
      printf ("\n Bucket capacity after output = %d",
                                   bucket capacity);
        time Limit ++;
    }
    else if (new Packet > capacity || (new Packet + bucket
                        capacity > capacity) {
          printf ("\n New packet cannot be added to
                        packet");
        bucket capacity -= output Rate;
    printf ("\n Bucket capacity after output = %d",
                                        bucket capacity);
        time limit ++;
    }
        bucket capacity = 0)
```

```
prin
ti
ex
}
}
}
```

→ Output

```
Enter
buck
Enter
New
buck
Enter
buck
Buck
Ente
Buck
```



```
printf ("\n Bucket capacity after output =%d",
                              bucket capacity);
timelimit ++;
exit (0);
}
}
}
```

→ Output:

```
Enter New packet size = 15
bucket capacity : 10
Enter new packet size = 27
New packet cannot be added to bucket
bucket capacity after output = 5
Enter new packet size = 0
bucket capacity : 5
bucket capacity after output : 0
Enter new packet size : 0
Bucket capacity after output : -5
```

3/8/2024