*from* time *import* time  
  
  
*class* TokenBucket(*object*):  
 *"""An implementation of the token bucket algorithm.  
  
 >bucket = TokenBucket(80, 0.5)  
 print bucket.consume(10)  
 True  
 print bucket.consume(90)  
 False  
 """  
  
 def \_\_init\_\_*(*self*, tokens, fill\_rate):  
 *"""tokens is the total tokens in the bucket. fill\_rate is the  
 rate in tokens/second that the bucket will be refilled."""  
 self*.capacity = *float*(tokens)  
 *self*.\_tokens = *float*(tokens)  
 *self*.fill\_rate = *float*(fill\_rate)  
 *self*.timestamp = time()  
  
 *def* consume(*self*, tokens):  
 *"""Consume tokens from the bucket. Returns True if there were  
 sufficient tokens otherwise False."""  
 if* tokens <= *self*.tokens:  
 *self*.\_tokens -= tokens  
 *else*:  
 *return False  
 return True  
  
 def* get\_tokens(*self*):  
 *if self*.\_tokens < *self*.capacity:  
 now = time()  
 delta = *self*.fill\_rate \* (now - *self*.timestamp)  
 *self*.\_tokens = *min*(*self*.capacity, *self*.\_tokens + delta)  
 *self*.timestamp = now  
 *return self*.\_tokens  
  
 tokens = *property*(get\_tokens)  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 *from* time *import* sleep  
  
 bucket = TokenBucket(80, 1)  
 *print*("tokens =", bucket.tokens)  
 *print*("consume(10) =", bucket.consume(10))  
 *print*("consume(10) =", bucket.consume(10))  
 sleep(1)  
 *print*("tokens =", bucket.tokens)  
 sleep(1)  
 *print*("tokens =", bucket.tokens)  
 *print*("consume(90) =", bucket.consume(90))  
 *print*("tokens =", bucket.tokens)

import java.io.\*;

import java.util.\*;

class LBA {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the token capacity of bucket");

int token=sc.nextInt();

System.out.println("Enter the burst length");

int burstLn=sc.nextInt();

System.out.println("");

int packets=10;

for (int i=1;i<=token;i++) {

System.out.println("Packet "+ packets-- +" with token number "+i);

}

}

}

package tockenbucket;

import java.util.Scanner;

public class Tockenbucket {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int sum=0;

System.out.println("Enter capacity of the bucket");

int C=sc.nextInt();

System.out.println("Enter the token generation speed");

int rho=sc.nextInt();

System.out.println("Enter the Outgoing speed");

int M=sc.nextInt();

System.out.println("Enter the number of users");

int num=sc.nextInt();

for(int i=0;i<num;i++){

System.out.println("enter the time");

int time=sc.nextInt();

System.out.println("enter the speed of sending");

int speed=sc.nextInt();

sum=sum+speed\*time;

}

int s=C/(M-rho)+1;

int data=s\*M;

if(data>sum){

System.out.println("finished into "+s+" seconds");

}

else{

sum=sum-data;

sum=sum/rho;

sum=sum+s;

System.out.println("finished into "+sum+" seconds");

}

}

}

Output:

run:

Enter capacity of the bucket

250

Enter the token generation speed

2

Enter the Outgoing speed

25

Enter the number of users

3

enter the time

50

enter the speed of sending

10

enter the time

10

enter the speed of sending

20

enter the time

10

enter the speed of sending

30

finished into 373 seconds

BUILD SUCCESSFUL (total time: 33 seconds)