

题目描述:

M ($1 \leq M \leq 20$) 辆车需要在一条不能超车的单行道到达终点, 起点到终点的距离为 N ($1 \leq N \leq 400$)。速度快的车追上前车后, 只能以前车的速度继续行驶, 求最后一辆车到达目的地花费的时间。

注: 每辆车固定间隔 1 小时出发, 比如第一辆车 0 时出发, 第二辆车 1 时出发, 依次类推

输入描述:

第一行两个数字: M N 分别代表车辆数和到终点的距离, 以空格分隔。

接下来 M 行, 每行 1 个数字 S, 代表每辆车的速度。 $0 < S < 30$

输出描述:

输出: 最后一辆车到达目的地花费的时间。

补充说明:

收起

示例 1

输入:

2 11

3

2

输出:

5.5

说明:

2 辆车, 距离 11, 0 时出发的车速度快, 1 时出发的车, 达到目的地花费 5.5

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        while (sc.hasNext()) {  
            String line = sc.nextLine();  
            String[] cnt = line.split(" ");  
            int count = Integer.parseInt(cnt[0]);  
            int distance = Integer.parseInt(cnt[1]);  
  
            String[] inputs = new String[count];  
            for (int i = 0; i < count; i++) {  
                inputs[i] = sc.nextLine();  
            }  
            processTime(count, distance, inputs);  
        }  
    }  
}
```

```
}
```

```
public static void processTime(int count, int distance, String[] inputs) {  
    float mintime = 0;  
    for (int i = 0; i < count; i++) {  
        int speed = Integer.parseInt(inputs[i]);  
        float curtime = (float) distance / speed;  
        if (i == 0) {  
            mintime = curtime;  
            continue;  
        }  
        if (curtime < mintime - 1) {  
            mintime -= 1;  
        } else {  
            mintime = curtime;  
        }  
    }  
    System.out.println(mintime);  
}  
}
```