Monday, January 3, 2022 11:26 PM

PRATA - Roti Prata

no tags

IEEE is having its AGM next week and the president wants to serve cheese prata after the meeting. The subcommittee members are asked to go to food connection and get P(P<=1000) pratas packed for the function. The stall has L cooks(L<=50) and each cook has a rank R(1<=R<=8). A cook with a rank R can cook 1 prata in the first R minutes 1 more prata in the next 2R minutes, 1 more prata in 3R minutes and so on(he can only cook a complete prata) (For example if a cook is ranked 2.. he will cook one prata in 2 minutes one more prata in the next 4 mins an one more in the next 6 minutes hence in total 12 minutes he cooks 3 pratas in 13 minutes also he can cook only 3 pratas as he does not have enough time for the 4th prata). The webmaster wants to know the minimum time to get the order done. Please write a program to help him out.

Input

The first line tells the number of test cases. Each test case consist of 2 lines. In the first line of the test case we have P the number of prata ordered. In the next line the first integer denotes the number of cooks L and L integers follow in the same line each denoting the rank of a cook.

Output

Print an integer which tells the number of minutes needed to get the order done.

Example

Input:

3

10

4 1 2 3 4

8

1 1

8

8 1 1 1 1 1 1 1 1

Output:

12

36

1

Submit solution!

From <https://www.spoj.com/problems/PRATA/>

```
class ParathaSolution {
    public int  minTimeForParatha(int order, int noOfCook, int ranks[]) {
        int low = 0;
        int high = Integer.MAX_VALUE;
        int result = Integer.MAX_VALUE;

        while (low <= high) {
            int mid = (low + high) / 2;
            int givenTime = mid;
            if(isParathaMakingPossible(givenTime, ranks, noOfCook, order))
            {
                result=Math.min(result, mid);
                high=mid-1;
            }
            else{
                low=mid+1;
            }

System.out.println(result);
        }
        return result;
    }
    private boolean isParathaMakingPossible(int givenTime,int[] ranks,
     int noOfCook, int order)
     {
         int taotalCount=0;
         int time =0;
         for (int i = 0; i < ranks.length; i++) {
         int count=2;
          time=ranks[i];

            while(time<=givenTime)
            {

               taotalCount++;
               time+=((count)*ranks[i]);
               count++;

            }

            if (taotalCount>=order) {
                return true;

            }

         }

     return false;
        }
}
```