

Johnny Listens to Music

Description

Johnny loves to listen to music so much. Each day he decides to add, delete or listen to a music track on his laptop by doing one of the following operations:

1. Download a new track.
2. Delete the last downloaded track that is not deleted already.
3. Listen to the shortest track that is already on his laptop.

Each test case Johnny starts with an empty collection of tracks, at day i Johnny decides which action to be taken depending on the value of $R[i] \% 3$. if $R[i] \% 3 = 0$ he listens to the shortest track on his laptop, if $R[i] \% 3 = 1$ he deletes the last downloaded track that is not deleted already, if $R[i] \% 3 = 2$ he downloads a new track whose length is $R[i]$ minutes. Your task is to compute the total number of minutes Johnny spends listening to music.

To keep the input small, it will be codified in the following way. You will be given an array h . Use the following pseudo-code on h to generate an array R .

- input array: h
- output array: R (of size n)
- $j := 0$
- $m := \text{size of } h$
- for $i := 0$ to $n-1$
 - $R[i] := h[j]$
 - $s := (j+1)\%m$
 - $h[j] := ((h[j] \wedge h[s]) + 13) \% 835454957$
 - $j := s$

This code, along with the constraints, ensures that the length of each track is between 0 and 835454956, inclusive. In the above code, $\%$ is the modulus operator and \wedge is the bitwise XOR binary operator. If x and y are integers, $(x \wedge y)$ represents the bitwise XOR operation on them in C/C++ and Java.

Input Format

Input will start with T number of test cases. Each test case will consist of 2 lines the first line starts with $0 < n \leq 10,000,000$ (size of array R), $0 < m \leq 50$ (size of array h). The second line will contain M numbers $0 \leq h[i] \leq 835454956$ which are the elements of array h .

Output Format

For each test case, output the result using the following format:

$k. S$

Where k is the test case number (starting at 1), a single period, a single space, and S is the total time in minutes spent by Johnny listening to music.

Sample Input	Sample Output
2	1. 5
6 6	2. 52
8 5 2 4 8 9	
10 4	

