Vending Machines (`vendingmachines`)

Carlo has bought T vending machines, each machine has N items, each with a price $P_1, P_2, \ldots, P_N$.

Each buyer can insert some money in the machine or buy an item. Each item can only be bought if there is enough money in the machine. After buying an item, the leftover money remains in the machine and can be used to buy another item.

Figure 1: Carlo's vending machines.

However, Carlo is worried than some machine might have been hacked! Luckily, Carlo has the history of the Q transactions that happened since the machine was turned on. Each transaction is represented by a signed integer L_i:

- if the transaction is positive (i.e. $+L_i$) then L_i euros were inserted in the machine;
- if the transaction is negative (i.e. $-L_i$) then item $abs(L_i)$ was bought.

If, at any point in time, the price of purchased items is greater than the inserted money then the machine was hacked. Help Carlo understand if his machines were hacked!

> ☞ Among the attachments of this task you may find a template file `vendingmachines.*` with a sample incomplete implementation.

Input

The first line contains the number of vending machines T. The next $3T$ lines describe each vending machines.

Each vending machine is represented by 3 lines:

- the first line contains the two integers N and Q;
- the second line contains the N integers $P_1, P_2, \ldots, P_N$;
- the third line contains the Q transactions $L_1, L_2, \ldots, L_Q$.

Output

You need to write a single line for each machine with the message "`HACKER`" if the machine was hacked otherwise "`OK`".

Constraints

- $1 \leq T \leq 100\,000$.
- $1 \leq N \leq 100\,000$.
- $1 \leq Q \leq 100\,000$.
- The sum of N and Q over all vending machines does not exceed $100\,000$.
- $0 \leq P_i \leq 10\,000$ for each $i = 1, 2, \ldots, N$.
- $-N \leq L_i \leq 10\,000, L_i \neq 0$ for each $i = 1, 2, \ldots, Q$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (30 points) All the money was inserted before buying the any item.

– **Subtask 3** (50 points) $T \leq 5$, $N, Q \leq 100$, $P_i, L_i \leq 100$.

– **Subtask 4** (20 points) No additional limitations.

Examples

input	output
1 2 4 3 4 +5 -1 +2 -2	OK
2 5 5 4 14 14 30 24 -4 +19 -5 +27 +27 10 10 22 14 29 23 22 23 26 6 22 2 +11 +27 +18 -6 +9 -8 -3 -10 +30 -4	HACKER OK

Explanation

In the **first sample case** there's a single machine. This machine has performed 4 transactions:

- 5 euros are inserted in the machine;
- item 1 is bought leaving the machine with 2 euros;
- 2 euros are inserted in the machine;
- item 2 is bought leaving the machine without money.

Every time an item has been bought there were enough money in the machine. The machine was not hacked so the answer is "`OK`".

In the **second sample case** there are two machines. The first machine is hacked at the first transaction, the hacker bought item 4 without inserting any money in the machine. The second machine has not been hacked.