Problem A. The Problems Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

*Eddard and his secret partner are betting on how many problems each contestant will solve in **Game Of Coders** this year. Eddard knows his secret partner would win the bet if they played fairly, so he decided to get some outside help – your help.*

m contestants have signed up for the contest, and will be given n problems and t minutes to solve them. Each contestant i has a rating r_i, and each problem i has a difficulty rating d_i, time to solve ts_i, and time to fail tf_i.

Eddard believes this is how each contestant c performs:

1. The contestant will pick a problem p uniformly at random from the problems they haven't picked before.
2. If their rating is greater than or equal to the problem's difficulty ($r_c \geq d_p$), they spend ts_p minutes solving the problem and succeed. Otherwise, they spend tf_p minutes trying to solve it before giving up. In both cases, they move on to another problem afterwards.

This process repeats until the time runs out or the contestant has picked every problem once.

For each contestant, print the expected number of solved problems after the contest ends.

Note that if the time runs out at the same minute a contestant finishes solving a problem, the problem is counted as solved.

Input

The first line of input consists of 3 integers $n, m, t (1 \leq n \leq 12, 1 \leq m \leq 10^5, 1 \leq t \leq 300)$

The next three lines of input describe the problems and consist of n integers each:

- $d_1, d_2, ..., d_n (1 \leq d_i \leq 9999)$ – The difficulties of the problems.
- $ts_1, ts_2, ..., ts_n (1 \leq ts_i \leq 300)$ – The time to solve each problem.
- $tf_1, tf_2, ..., tf_n (1 \leq tf_i \leq 300)$ – The time to fail at each problem.

The last line consists of m integers $r_1, r_2, ..., r_m (1 \leq r_i \leq 9999)$, The rating of each contestant.

Output

print m integers $e_1, e_2, ..., e_m$, where e_i is the expected number of problems contestant i will solve **modulo** $10^9 + 7$**.**

Formally, the expected number can be expressed as an irreducible fraction $\frac{x}{y}$. You have to print the value of $x \cdot y^{-1} \bmod 10^9 + 7$, where y^{-1} is an integer such that $y \cdot y^{-1} \equiv 1 \pmod{10^9 + 7}$.

Examples

standard input	standard output
1 2 50 1000 50 50 500 1500	0 1
6 8 150 800 1200 1500 1800 2200 2600 2 5 15 25 40 60 10 12 20 40 30 100 900 1300 3000 1260 2500 1750 200 1979	433333337 100000002 6 100000002 716666675 766666674 0 21

Note

In the first test Case:

the first contestant has rating 500, they pick the only problem with difficulty 1000 and spend the 50 minutes of the contest trying to solve it but fail, thus solving 0 problems.

The second contestant has rating 1500, so they're able to solve it in 50 minutes, thus solving 1 problem.

In the second test case, the answer as irreducible fractions is:

$\frac{19}{30}, \frac{13}{10}, \frac{6}{1}, \frac{13}{10}, \frac{199}{60}, \frac{59}{30}, \frac{0}{1}, \frac{169}{60}$

Problem B. Re-Indexing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

A book's table of contents is a page which lists every chapter and the number of its first page and chapters are sorted by their first page. For example, the following is a table of contents:

- Chapter1 – 1
- Chapter2 – 43
- Chapter3 – 60

While reading, Eddard dropped his book and its table of contents got shuffled, the rest of the book is fine, and every chapter is still mapped to its first page correctly in the table of contents, but the order of the chapters in the list got messed up. For example, the above table of contents might now look like this:

- Chapter2 – 43
- Chapter3 – 60
- Chapter1 – 1

Now, Eddard can't know which chapter he is supposed to read, he only remembers the title of the last chapter he finished. So, he went to his wise secret partner to ask for help, but his partner said he should act responsibly and take care of it himself.

Since Eddard obviously can't take care of it himself, he needs your help. Also, be ready to be asked many times as he will ask you repeatedly for help in the same book.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n, q ($1 \le n, q \le 10^5$) – The number of chapters in the book and the number of queries, respectively.

Each of the next n lines contains a string s and an integer p ($1 \le |s| \le 15, 1 \le p \le 10^9$) – the title of the chapter and its starting page. It is guaranteed that each chapter is at least one page (No two chapters in the table of contents will have the same starting page) and that no two chapters have the same name.

Each of the next q lines contains a string c ($1 \le |c| \le 15$) – the title of the last chapter Eddard just finished. It is guaranteed that the title of the chapter exists in the table of contents.

It is guaranteed that sum of n over all test cases does not exceed 10^5. All strings consist of lowercase and uppercase letters and numbers.

Output

For each query, print the title of the next chapter Eddard is supposed to read in a single line. If the chapter was the last one in the book, print "No More Chapters".

Example

standard input	standard output
2 3 3 Chapter2 43 Chapter3 60 Chapter1 1 Chapter1 Chapter2 Chapter3 3 1 SecondChapter 4 FirstChapter 1 ThirdChapter 24 FirstChapter	Chapter2 Chapter3 No More Chapters SecondChapter

Note

The first test case is explained in the statement

In the second test case, the table of contents should be ordered as follows:

- FirstChapter – 1
- SecondChapter – 4
- ThirdChapter – 24

And the next chapter to FirstChapter should be SecondChapter

Problem C. The Rectangular City

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Eddard and his secret partner are visiting The Rectangular City for k days, and they have a plan P for their trip.

The Rectangular City is built as a nm grid of blocks. Each block (x, y) is of type $t_{x,y}$ and has a teleportation device and a metro station.

On the i-th day, the couple will visit any block of their choice of type P_i.

For example, for $P =$ "goc", they will spend 3 days. On the first day, they'll visit any block of type **g**, the next day, a block of type **o**, and on the last day, a block of type **c**.

On the first day, they arrive at any block they want. Afterwards, going from one block to another is done in the following steps:

1. **First**, they can use the teleportation device to arrive at any block of the same type as the current block they're in.
2. **Then,** they can take the metro to travel to any block in the city.

The teleportation device is free. However, the metro isn't. If they use the metro to travel from block (x_1, y_1) to block (x_2, y_2), they pay $|x_1 - x_2| + |y_1 - y_2|$ rectangles (The Rectangular City's currency). Check the examples and explanations below.

Since Eddard and his secret partner aren't very rich, they want to spend the least amount of rectangles. Find the minimum amount of rectangles they need to pay in their trip.

Input

The first line of input consists of a single integer $T (1 \le T \le 10^4)$ – the number of test cases.

Each test case is given as the following:

The first line consists of 3 integers $n, m, k (1 \le n, m \le 10^6, 1 \le k \le 10^6)$.

Each of the next n lines consists of a string t_x of size m, where $t_{x,y}$ is the type of block (x, y).

The next line consists of a string P of size k – the plan for the trip. It is guaranteed that each character in P appears at least once in t.

All strings consist of lowercase letters.

It is guaranteed that both the sum of $n * m$ and the sum of k over all test cases do not exceed 10^6.

Output

For each test case, print a single integer, the minimum amount of rectangles they need to pay in their trip.

Example

standard input	standard output
3 3 4 5 anac nync anac ycnan 5 5 4 abbbb ccccc ddddd eeeee ffffg adga 1 1 2 x xx	5 12 0

Note

A possible scenario in the first test case is the following:

- Start at block (2, 2) of type **y**.
- Skip the teleportation step and use the metro to go to block (2, 4) of type **c**, for 2 rectangles.
- Use the metro again to (2, 3) of type **n** for 1 rectangle.
- Use the metro to (1, 3) of type **a** for 1 rectangle.
- Use the metro to (1, 2) of type **n** for 1 rectangle.

The Total cost is $2 + 1 + 1 + 1 = 5$ rectangles.
For the second test case:

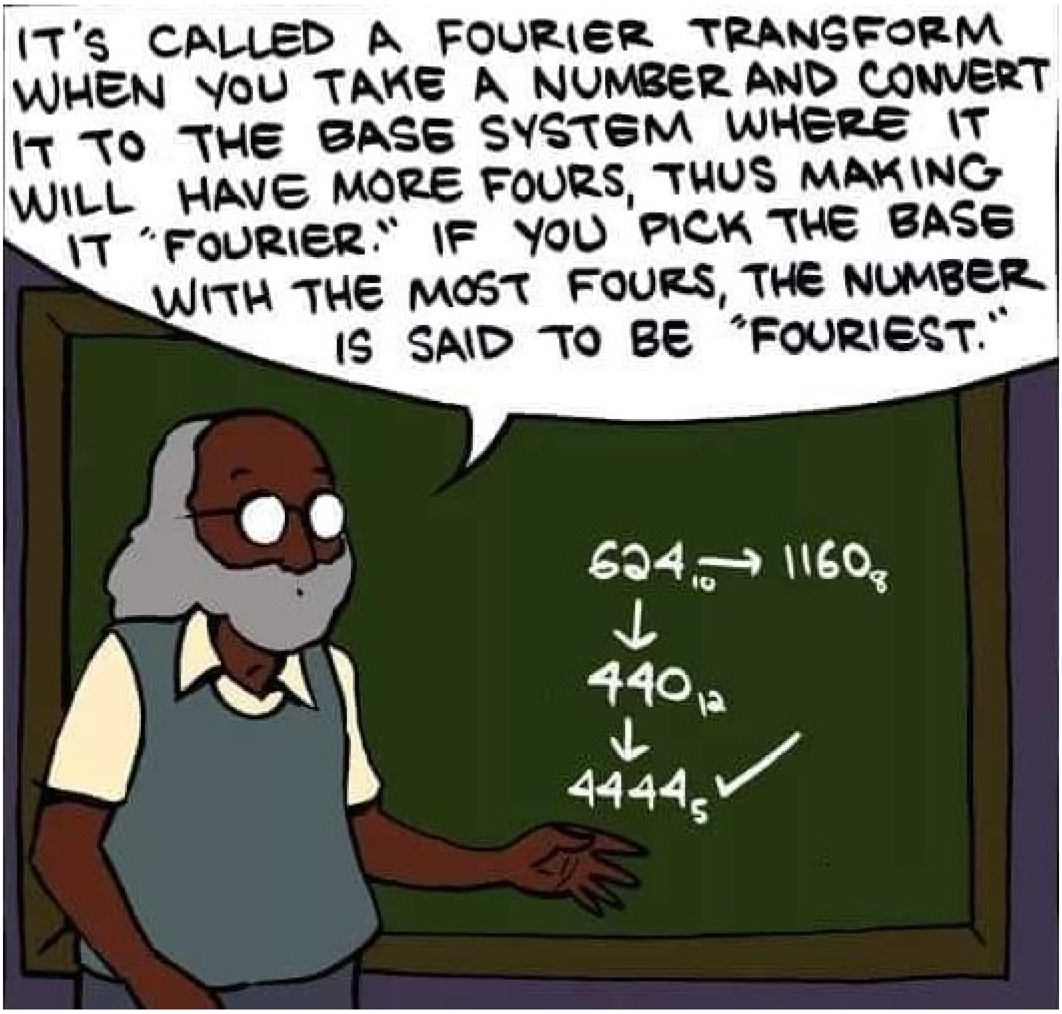
- Start at block (1, 1).
- Metro to (3, 1) for 2 rectangles.
- Teleport to (3, 5) then metro to (5, 5) for 2 rectangles.
- Metro to (1, 1) for 8 rectangles.

The total cost is $2 + 2 + 8 = 12$ rectangles.

Problem D. The FFT Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Eddard has wanted to learn about FFT for a very long time, but he's a little lazy and is intimidated by its difficulty and complexity. However, he found a simple meme explaining what the Fourier Transform is.



Fourier Transform explanation. (Check the notes if the image is not loading.

He decided to find a fast algorithm that can find the base for any number that makes it the "Fouriest", basically inventing the Fast Fourier Transform. Can you help him?

Input

The first line of input consists of a single integer $T(1 \leq T \leq 10)$, the number of test cases.

Each test case is given in one line containing a single integer $n(5 \leq n \leq 10^{12})$

Output

For each test case, print a single integer $b(2 \leq b \leq 2 \cdot 10^{12})$, the base where n becomes "Fouriest". If no b exists where the representation of n has at least 1 four, print -1. **If multiple solutions exist, print the largest.**

Example

standard input	standard output
4 10 24 20 44444	6 5 16 10

Note

Test Cases Explanation:

$(10)_{10} = (14)_6$

$(24)_{10} = (44)_5$

$(20)_{10} = (40)_5 = (14)_{16}$ Here, 20 produces 1 four in both bases 5 and 16, but 16 is greater.

44444 is fouriest in base 10

Image description:

It's called a Fourier Transform when you take a number and convert it to the base system where it will have more fours, thus making it "Fourier". If you pick the base with the most fours, the number is said to be "Fouriest".

Problem E. Tim Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

*Since watching the game theory session, the secret partner has been beating Eddard in Nim Game every time. So, Eddard got bored and decided they should spice it up a little bit and make a new game with his favorite topic, trees! He decided to call the new game **Tim Game**, short for Tree Nim Game. He really doesn't know how to come up with creative names.*

Eddard and his secret partner are playing a game where they are given a tree with n nodes rooted at node 1. The secret partner goes first, then they alternate turns.

Each turn, a player picks a node u **other than the root**, and removes it and all its descendants.

The last player to make a move wins. Equivalently, the player who can't make a move on their turn loses.

Can you find out who will win if both play optimally?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 4 \cdot 10^5$) – The size of the tree.

Each of the next $n-1$ lines describes an edge (u, v) ($1 \le u, v \le n$).

It is guaranteed that the given edges in each test case form a tree, and the sum of n over all test cases does not exceed $4 \cdot 10^5$.

Output

For each test case, print "The Secret Partner" if the secret partner wins. Otherwise, print "Eddard".

Example

standard input	standard output
3 6 1 2 2 3 2 4 1 5 5 6 10 1 2 2 3 3 4 4 5 3 6 1 7 7 8 7 9 9 10 2 1 2	The Secret Partner The Secret Partner The Secret Partner

Note

In the first test case, one possible scenario is the following:

- The secret partner picks 3 and removes it.
- Eddard picks 2 and removes it and its descendants (4).
- The secret partner picks 5 and removes it and its descendants (6).
- Eddard can't make a move as only the root remains and loses.

In the second test case, one possible scenario is the following:

- The secret partner picks 6 and removes it.
- Eddard picks 3 and removes it and its descendants $(4, 5)$.
- The secret partner picks 10 and removes it.
- Eddard picks 8 and removes it.
- The secret partner picks 9 and removes it.
- Eddard picks 2 and removes it.
- The secret partner picks 7 and removes it.
- Eddard can't make a move as only the root remains and loses, again.

No need to explain the third test case, The Secret Partner always wins anyway.

Problem F. Fibonacci Strings

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Eddard likes strings. His partner likes Fibonacci numbers. Thus, The Fibonacci Strings!

The Fibonacci string sequence F can be defined given F_1, F_2 recursively as follows:

$$F_i = F_{i-1} + F_{i-2} \text{ for } i > 2$$

For example, for F_1 =“a”, F_2 = “b”; F = “a”, “b”, “ba”, “bab”, “babba”, “babbabab”, ..

Given F_1, F_2, and two integers x, k; find the k-th character of F_x.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains 2 integers n, m ($1 \le n, m \le 10^5$)

The second and third lines of each test case contain two strings of lowercase letters, F_1 of size n and F_2 of size m.

The fourth line of each test case contains 2 integers x, k ($1 \le x \le 1000, 1 \le k \le \min(10^{18}, |F_x|)$)

It is guaranteed that the sum of $n + m$ over all test cases does not exceed 10^5

Output

For each test case, output a single character, the k-th character in F_x.

Example

standard input

```
5
1 1
a
b
6 7
2 3
ab
xyz
1 2
4 4
nnnn
nnnn
1000 1000000000000000000
2 2
ab
cd
10 10
7 6
partner
secret
3 11
```

standard output

```
a
b
n
b
n
```

Note

In the first test case, $F_6 =$ “babbabab”, its 7th character is “a”.

In the second test case, $F_1 =$ “ab”, its 2nd character is “b”.

In the third test case, all strings consist only of the character “n”

Problem G. Symmetric Subarrays

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Eddard and his secret partner believe arrays should be symmetric, and any asymmetric array is worthless. So, they defined a function $V(a)$ as the value of array a as follows:

- If the array is symmetric*, $V(a)$ is equal to the sum of all elements in a.
- Otherwise, $V(a) = 0$.

Given an array a, calculate $\sum_{l=1}^{n} \sum_{r=l}^{n} V(a[l..r])$, where $a[l..r]$ is the subarray of a from index l to r.

Since the answer can be too large, print it modulo $10^9 + 7$.

* A symmetric array is an array that stays the same when reversed. More formally, in a symmetric array a of n integers, $a_i = a_{n+1-i}$ for $1 \le i \le n$. For example, the arrays $[1, 3, 1], [5, 9, 9, 5], [25]$ are symmetric, while $[1, 2], [4, 6, 4, 4]$ are not.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains the array length n ($1 \le n \le 10^6$).

The second line of each test case contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, print the answer modulo $10^9 + 7$.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>5
3
1 2 1
4
8 9 9 8
5
18 5 18 7 29
2
1 2
5
100 100 100 100 100</pre></td><td><pre>8
86
118
3
3500</pre></td></tr>
</table>

Note

In the first test case, the subarrays of $[1, 2, 1]$ are $\{[1], [2], [1], [1, 2], [2, 1], [1, 2, 1]\}$.

Only $[1], [2], [1], [1, 2, 1]$ are symmetric, so the answer is their sum: $(1) + (2) + (1) + (1 + 2 + 1) = 8$.

Problem H. Hot Cappuccino

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Eddard's secret partner likes matrices, so they went on a trip to The Matrix City, which is famous for its coffee shops that make the best cappuccino and hot chocolate. While walking around the city, however, they lost each other. Can you help them meet again?

The city consists of $n*m$ blocks, each block (x, y) with a coffee shop that serves cappuccino, hot chocolate, both, or neither.

Initially, Eddard is at block $(1, 1)$ while his partner is at (n, m). Each of them can move in the four directions. Since it's already summer and the weather is absolutely terrible, walking is tiresome and each step costs 1 willpower.

There is, however, something that can save some willpower for them...

Eddard likes hot chocolate, and his partner likes cappuccino. If they're currently at a block with a coffee shop that serves their preferred drink, they can drink and move to any adjacent block without spending willpower.

Find the minimum sum of willpower spent by both so that they can meet at any block.

Input

The first line of input consists of one integer $T(1 \le T \le 10^4)$ – The number of test cases.

The first line of each test case consists of 2 integers $n, m(1 \le n, m \le 10^6, n \cdot m \le 10^6)$

The i-th line of the next n lines consists of m integers $a_{i,1}, a_{i,2}, ..., a_{i,m}$, where $a_{i,j}$ describes the drinks served at the coffee shop in block (i, j):

- 0 for neither cappuccino nor hot chocolate
- 1 for cappuccino
- 2 for hot chocolate
- 3 for both

It is guaranteed that the sum of $n \cdot m$ over all test cases will not exceed 10^6

Output

Print the answer for each test case on a single line.

Example

standard input	standard output
4 4 6 2 2 0 0 0 0 0 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 10 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 3 3 3 3 3 3 1 1 0	2 10 0 0

Note

In the first test case:

Eddard can move as follows: $(1, 1) >^0 (1, 2) >^0 (2, 2) >^0 (2, 3) >^1 (3, 3)$

His partner can move as follows: $(4, 6) >^0 (4, 5) >^0 (3, 5) >^0 (3, 4) >^1 (3, 3)$

Problem I. The Vampire Partner

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Don't worry, I'm not a vampire.

— *The Secret Partner*

Eddard suspects his partner is a vampire. Can you help him find out the truth?

Everyone knows only vampires drink cappuccino, so Eddard decided to do a little trick. He would leave some cappuccino on a table where his partner is, then come back and check if his partner drank (or added) any.

Eddard put n cups of cappuccino on the table and measured the amount of cappuccino in every cup, but he only had a small piece of paper, so he couldn't write down how much each cup had. Instead, he could only write down one number. So, he decided to encode the information into one number using a function.

He defined $F(a) = \sum_{i=1}^{n}(i * a_i)$ where a is an array of n integers. He then wrote down the result of $F(c)$ on the piece of paper, where c_i is the amount of cappuccino in the i-th cup.

Eddard left the cups at the table and came back after a while and measured the cups again and found that the amount of cappuccino in the i-th cup is now p_i. Can you help Eddard find out if the amount of cappuccino **possibly** stayed the same?

Assume that neither the number of cups nor their positions changed.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains two integers $n, F(c)$ ($1 \le n \le 10^6, 0 \le F(c) \le 10^{18}$).

The second line of each test case contains n integers $p_1, p_2, ..., p_n$ ($0 \le p_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, if the amount of cappuccino in the cups has **certainly** changed, print "NO". Otherwise, print "YES".

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as a positive answer).

Example

standard input	standard output
5 5 30 2 4 2 1 2 4 100 10 10 10 10 1 5 4 3 20 1 4 1 1 0 0	YES YES NO NO YES

Note

Remember to use fast I/O.

In the first test case, $F(p) = F(c) = 30$. Thus, it is possible that the amount of cappuccino has not changed.

In the third test case, however, $F(p) = 4$, which differs from $F(c)$. So, it is certain that the amount of cappuccino changed.

Note that in the first test case, it is possible that the amount of cappuccino has changed. For example, c could be equal to $\{2, 2, 2, 2, 2\}$, which would still lead to $F(p) = F(c)$. The answer is still "YES" as the question is about the possibility of the cups not changing.

Problem J. Just One More Bro, I Swear

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Eddard is preparing the Game of Coders problemset, but he can't stop adding hard problems to the contest. His secret partner keeps begging him to make it a little easier, but he gets excited whenever he finds an idea for a hard problem, and he can't find any interesting ideas for easy ones. So his partner decided to take matters into their own hands and make this problem.

Given the number of problems in a contest Eddard will prepare, what is the expected number of hard problems?

Input

The input consists of a single integer n ($1 \le n \le 15$) – the number of problems.

Output

Print a single integer, the expected number of hard problems.

Examples

standard input	standard output
12	12
10	10
15	15
3	3

Note

This is not a trick, just print the input back :3

Problem K. The Red Tomato

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Eddard and his secret partner found a really cute song called **"I am the red tomato"** and they really loved it, but something piqued their interest..

"I am the red tomato, planted among the vegetables. You eat me until you're full, and your cheeks go red."

The mystery here is.. how much tomato can they eat?

Eddard and his partner have a hypothesis: The maximum total weight of tomatoes they can eat at one sitting is w grams, regardless of the number of tomatoes. For example, if $w = 9$, they can eat 3 tomatoes of weights $6, 2, 1$, as their total weight is 9, but they can't eat 2 tomatoes of weights $7, 4$, as their total weight is 11.

The goal is to find the value w. So, they decided to run the following experiment e times:

They put n tomatoes of weights $a_1, a_2, ..., a_n$ on a table. Then, they start eating the tomatoes from left (1) to right (n) until they can't eat any more tomatoes. After they're done, they write down the number of tomatoes left m.

For example, if $w = 5$, $a = [3, 4, 2]$, they will eat the first tomato of weight 3 then stop since they can't eat the second tomato, as the total weight would be 7. They will, then, write down $m = 2$ as the number of remaining tomatoes.

Can you help them find the value w after running multiple experiments?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer e ($1 \le e \le 10^6$) – The number of experiments they are running. The description of the experiments follows.

The first line of each experiment contains 2 integers n, m ($1 \le n \le 10^6, 0 \le m \le n$) – The number of tomatoes at the start and the end of the experiment, respectively.

The second line of each experiment contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$) – The weights of the tomatoes.

It is guaranteed that sum of n over all test cases does not exceed 10^6

Output

For each test case, print a single integer w ($0 \le w \le 10^{18}$) – the answer.

If multiple answers exist, print the smallest. if no answer exists, print "False Hypothesis".

Example

standard input	standard output
4 3 1 0 5 3 2 3 4 2 5 1 1 2 1 1 1 2 1 0 5 1 1 6 1 1 1 100 2 3 0 3 5 6 4 1 1 4 2 4	5 5 0 False Hypothesis

Note

In the first test case, 3 experiments are run:

- In the first experiment, $a = [5]$, $m = 0$, which means they ate the only tomato with weight 5.
- In the second experiment, $a = [3, 4, 2]$, $m = 2$, which means they ate only the first tomato with weight 3 and couldn't eat anymore.
- In the third experiment, $a = [1, 2, 1, 1, 1]$, $m = 1$, which means they ate the first 4 tomatoes with a total weight of 5, but couldn't eat the last tomato.

From these 3 experiments, it can be deduced that the smallest possible value for w is 5

In the fourth test case, after running the 2 experiments, it can be deduced that no value w exists that satisfies both experiments. Thus, the hypothesis is false.

Problem L. Growing Letters

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Every day, we grow, learn, and overcome challenges.

— *The Secret Partner*

Just like us, letters grow too.

Define $F_0 =$ "a"; $F_i = F_{i-1} +$ "b" $+ F_{i-1}$ for all $i \geq 1$.

For example, $F_1 = F_0 +$ "b" $+ F_0 =$ "aba".

You are given a string S, but since it can be very large, it is described as an array a of n integers, where $S = F_{a_1} + F_{a_2} + ... + F_{a_n}$.

For example, for $a = \{1, 0, 0\}, S =$ "aba" + "a" + "a" = "abaaa".

The letters in S merge and grow together. This is done as follows:

1. Find the least i such that $S_i = S_{i+1}$. If no such i exists, the growing process is complete.
2. S_i is set to the next letter of the alphabet. In case $S_i =$ 'z', It is set to 'a'.
3. S_{i+1} is erased.
4. Repeat.

For example, for $S =$ "abaaaaba", the growing process goes as follows:

ab(aa)aaba → ab(b)aaba

a(bb)aaba → a(c)aaba

ac(aa)ba → ac(b)ba

ac(bb)a → ac(c)a

a(cc)a → a(d)a

Can you find the size of S after the growing process is complete?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$) – The size of the array.

The second line of each test case contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 15$) – The elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed 10^5

Output

For each test case, output a single integer, the size of S after the growing process.

Example

standard input	standard output
3 4 1 0 0 1 8 0 0 0 0 0 0 0 0 3 0 3 0	3 1 13

Note

The first test case is described in the example in the statement.

In the second test case, $S =$ "aaaaaaaa". It grows into "d".

Problem M. Maximum Subarray Alternating Sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The secret partner got tired of the difficult problems Eddard kept adding to the contest, and decided to take matters into their own hands once again and add an easy classical problem, the maximum subarray sum: "Given an array a of n integers, find the maximum sum of any subarray."

But, of course, Eddard decided to sabotage the problem and make it a little harder. After all, why shouldn't he? Now, the problem is the maximum subarray **alternating** sum.

The alternating sum S of an array b is given as follows:

$S(b) = b_1 - b_2 + b_3 - ... + (-1)^{n+1} \cdot b_n$.

In other words, $S(b) =$ sum of elements with odd indices $-$ sum of elements with even indices.

For example, for $a = [3, 7, 1, 5]$:

$S(a_{2..4}) = a_2 - a_3 + a_4 = 7 - 1 + 5 = 11$, which is also the maximum subarray alternating sum.

$S(a_{1..2}) = a_1 - a_2 = 3 - 7 = -4$.

Can you find the maximum subarray alternating sum of the array?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^6$) – The size of the array.

The second line of each test case contains n integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10^9$) – The elements of the array.

Output

For each test case, print a single integer, the maximum subarray alternating sum of the array a.

Example

standard input	standard output
3 4 3 7 1 5 5 1 0 1 0 1 3 1 4 24	11 3 24