Graduation photo

Description

The members in VCN Research Center will graduate next year. At that time, they will stand in a line and take a graduation photo. Now, Hong starts to wonder in which way they arrange their standing positions can make the photo look best. Don't be worried, You may think the task for you is to use a strange evaluation function to find the best photo, the answer is NO. Because Hong is so powerful, he just needs you to output all the possible arrangements of the standing positions, and Hong will pick up the best one from them. There are N people in VCN Research Center, and each person can be represented in a lower case letter with no duplication. You should output the all the possible arrangements in the lexicographical order.

Lexicographical order is defined in following way. When we compare strings s and t, first we find the leftmost position with differing characters: $s_i \neq t_i$. If there is no such position (i. e. s is a prefix of t or vice versa) the shortest string is less. Otherwise, we compare characters s_i and t_i according to their order in alphabet.

Input

The first line contains an integer T, $1 <= T <= 10$, indicating the number of test cases.

For each test case, there are two lines. The first line contains an integer N, $1 <= N <= 6$, indicating the number of people. The second line contains a string length of N, only consisting of lower case letters, indicating the the letters representing the people.

Output

For each test case, output all possible standing arrangements in lexicographical order. Each arrangement should be on a line and represented as a string of length N.

Sample input

```
2
3
lyh
2
vc
```

Sample output

```
hly
hyl
lhy
lyh
yhl
ylh
cv
vc
```

Sigma sigma

Constrain

`1s, 256M`

Description

The pronunciations of σ and Σ are exactly the same, but it would be a challenge when we combine them together.

Denote that $\sigma_k(n) = \sum_{d|n} d^k$, i.e. the summation k-th power of all the divisors of n.

Find

$$S_k^m(n) = \begin{cases} \sigma_k(n) & , m = 0 \\ \sum_{i=1}^{n} S_k^{m-1}(i) & , m \geq 1 \end{cases}$$

Input

One line contains three space separated integers m, k and n.

$1 \leq m \leq 4, 1 \leq k \leq 4, 1 \leq n \leq 10^{11}$.

Output

Print the corresponding value of $S_k^m(n) \mod 1\,000\,000\,007$.

Sample input 1

```
2 2 2
```

Sample output 1

```
7
```

Sample input 2

```
3 3 3
```

Sample output 2

```
61
```

Hint

$$S_2^2(2)=\sum_{i=1}^{2}\sum_{j=1}^{i}\sigma_2(j)=\sigma_2(1)+\sigma_2(1)+\sigma_2(2)=1+1+5=7$$

$$\begin{aligned}
&S_3^3(3)=\sum_{i=1}^{3}\sum_{j=1}^{i}\sum_{l=1}^{j}\sigma_3(l)\\
=&\sigma_3(1)+\sigma_3(1)+\sigma_3(1)+\sigma_3(2)+\sigma_3(1)\\
+&\sigma_3(1)+\sigma_3(2)+\sigma_3(1)+\sigma_3(2)+\sigma_3(3)\\
=&1+1+1+9+1+1+9+1+9+28=61
\end{aligned}$$

Lanran's Prowssad

Lanran hates to create new password, because he often forgets the password after a period of time. Thus, Lanran comes up with an idea to construct new password using some rules. The idea is simple: Lanran remembers an original string S of length N. To create a new password P, he just needs to select a continuous substring of length M from the original string, and reverse this substring. That means, if the substring of length M starts from the k-th character in S, then

$$\forall\, i \in [k, k+M-1], S[i] = P[2k+M-i-1]$$
$$\forall\, i \in [1, k-1] \cup [k+M, N], S[i] = P[i]$$

Apparently, Lanran can generate at most $N-M+1$ different new passwords by reversing a substring of the original string S. One day, after Lanran has generated all $N-M+1$ passwords, he unfortunately forgets the original string S. Given all possible new passwords, can you try to recover S?

Input

The first line contains two integers N and M, $15 <= N <= 2000, 1 <= M <= 5$.

The second line contain all possible new passwords separated by a space. Each password has a length of N. The passwords may contain upper letters, lower letters and digits. Any new password generated by reversing a substring will appear exactly once.

Output

The output contains a string of length N, indicating the original string S.

Example Input 1

```
20 5
lalaarnalnnloveyeeye lalanlanranloveyeeye lalanllnarnaoveyeeye lalanlanranleeyevoye
lalanlanrevolnayeeye lalrnalnaanloveyeeye lalanlaolnarnveyeeye lalanlanraneyevoleye
lnalalanranloveyeeye lalnalanranloveyeeye lanalnalranloveyeeye lalanlanrayevolneeye
lalanlanvolnareyeeye lalannarnalloveyeeye lalanlanranloyeeyeve
```

Example Output 1

```
lalanlanranloveyeeye
```

No zeroes

Constrain

`1s, 64M`

Description

Denote $G(n) = n!$ without the tailing zeroes.

For example, $14! = 87178291200$, $G(14) = 871782912$.

Find the last k digits of $G(n)$.

Input

The first line will be a integer T, represents the test cases.

Each line of the following T lines contains two space separated integer k and n.

$0 \leq T < 4 \times 10^4, 1 \leq k \leq 8, 1 \leq n \leq 10^{18}$.

Output

For each test case, output one line for the last k digits of $G(n)$. If $G(n)$ contains less than k digits, output $G(n)$ instead. If last k digits of $G(n)$ starts with 0, you must print the leading zeroes.

Sample input

```
3
3 7
4 4
4 30
```

Sample output

```
504
24
0848
```

Lanrax

One day, the people in the VCN Research Center wrote a paper, but VinceBlack did not want to use any existing reference format. Therefore, he came up with a new reference format called **LANRAX**. Now he wishes you to write a LANRAX format generator.

We describe the LANRAX Reference Format in the way of strings and optional strings. A string is represented as `{StringName}`, and it can be regarded as a string variable. Also, a string can be directly expressed by a combination of some strings, like `<{String1}: {String2}, {String3}>`, which means `string1 + ": " + string2 + ", " + string3` where `+` is string concatenation. An optional string is in the format `[content]` or `~[content]` or `[content]~`, and it outputs its content only when all of its neighbor strings (the front one and back one) without `~` on its side are not null. That means, for `[]`, it demands both the front one and back one to be not null for content output. For `~[]`, it demands the back one to be not null to output, and the front one should be not null for `[]~`. **A string with zero length is null**.

For example:

`{NAME1} = "YEE", {NAME2} = "172", {NAME3} = null, {NAME4} = "is here"`

`{NAME1}[-]{NAME2} = "YEE-172"`

`{NAME1}[-]{NAME3} = "YEE"`

`{NAME1}[-]~{NAME3} = "YEE-"`

`{NAME3}[-]~{NAME2} = "172"`

`<Mr. {NAME1}[-]{NAME2}>[ ]{NAME4} = "Mr. YEE-172 is here"`

`<{NAME3}[-]{NAME3}>[ ]{NAME4} = null`

`{NAME1}[-]{NAME3}[+]{NAME4} = "YEEis here"` // Please notice that only the **nearest neighbor one** string could affect the output of `[]`.

For LANRAX, the final output should be `{LANRAX}` , which equals to `<@ {AUTHOR}; {PUBLISHER}; {TITLE}; {CITATION}; {TIME}.>`.

Please read the following rules really really carefully! Any different character (including a space) between your answer and the true answer will cause a wrong answer.

AUTHOR

- There can be multiple authors. The author names are `{Author1}`, `{Author2}`, ..., `{Author[x]}`, `[x]` is the number of the authors.
    - The name of author should be presented in this format `{Initials}[ ]{Author[x]FamilyName}`. The `{Initials}` is the combination of first letters of the words in the `{Author[x]FirstName}` with a dot behind, and they are separated by a space. For example, if the family name is `Black` and the first name is `Vince`, the result should be `V. Black`. If the first name is `William Jafferson` and the family name is `Clinton`, the result should be `W. J. Clinton`.

- If there is no author, `{AUTHOR}` is null.
- If there is one author, `{AUTHOR}` should be `{Author1}`.
- If there are two authors, `{AUTHOR}` should be `{Author1} and {Author2}`
- If there are three to five authors, `{AUTHOR}` should be like `{Author1}, {Author2}, {Author3} and {Author4}` (Similar format for 3 and 5 authors).
- If there are more than five authors, `{AUTHOR}` should be `{Author1} et al.`
- If there is any word in `{Author[x]FamilyName}` and `Author[x]FirstName` starting from lower letter, convert it to upper letter.

PUBLISHER

- The `{PUBLISHER}` should be `<<{City}[, ]{Province}>[, ]{Country}>[: ]{Publisher}`.
- For example:

  `{City} = "Shenzhen", {Province} = "Guangdong", {Country} = "China", {Publisher} = "SUSTech"`

  `{PUBLISHER} = "Shenzhen, Guangdong, China: SUSTech"`

  Another example:

  `{City} = null, {Province} = null, {Country} = null, {Publisher} = "Moon Station"`

  `{PUBLISHER} = "Moon Station"`

TITLE

- The `{TITLE}` should be `~["]{Title}["]~`.

CITATION

- If `{Type} = "BOOK"`, `{CITATION}` should be

  `<~[From ]{BookName}[~[, ]{Pages}]~>[~[ written by ]{BookAuthor}]~`.

  The string `{Pages} = <~[page ]{PageNumbers}>`, and `{PageNumbers}` derives from an array of integers in `{PageNumberArray}` which are the page numbers included. If there is only one page in `{PageNumberArray}`, then `{PageNumbers}` is equal to that page number. If there are multiple page numbers in `{PageNumberArray}`, `{PageNumbers}` should be all page numbers in increasing order separated by a semicolon and a space, and the last one should be separated by `and` (similar with authors). The continuous pages (more than 1 page) should be combined into a range, which is in the format of `start-end`.

  - Example 1:
    - Page numbers are [1,2,4,5,6,9,8,10,15].
    - `{Pages} = "page 1-2, 4-6, 8-10 and 15"`.
  - Example 2:
    - Page number is [4, 6]
    - `{Pages} = "page 4 and 6"` .
  - Example 3:
    - Page number is [1,2,3]

    - `{Pages} = "page 1-3"`.
- If `{Type} = "CONFERENCE"`, `{CITATION}` should be `<~[In ]{ConferenceName}>[~[ held in ]{ConferencePlace}]~`.
- If `{Type} = "ONLINE"`, `{CITATION}` should be

  `<~[Accessed online from ]{WebsiteName}>[~[, link:]{WebHttpLink}]~`.
- Please notice that `{BookAuthor}` is not represented in the rule of `{Author}`, but just as the raw input text.

TIME

- `{TIME} = <~[{Date}[ ]~]{MonthAbbr}>[ ]{Year}`, where the `{MonthAbbr}` is converted from `{Month}` through the following table.

{Month}	{MonthAbbr}
1	Jan.
2	Feb.
3	Mar.
4	Apr.
5	May
6	Jun.
7	Jul.
8	Aug.
9	Sep.
10	Oct.
11	Nov.
12	Dec.

Input

The input format is as following:

The first line is an integer n indicating the number lines following.

On each following line, the input format is `"String name":"String content"`. All possible `String name` and corresponding content formats are listed as follows.

- `"Author[x]FirstName":string` containing one or multiple words in alphabet separated by a space, length<20

- `"Author[x]FamilyName":string` containing one or multiple words in alphabet separated by a space, length<20
- `"City":string` containing one or multiple words in alphabet, length<20
- `"Province":string` containing one or multiple words in alphabet, length<20
- `"Country":string` containing one or multiple words in alphabet, length<20
- `"Publisher":string` containing one or multiple words in alphabet, length<20
- `"Title":string` containing one or multiple words in alphabet, digital number length<60
- `"Type":enum` can only be "BOOK", "CONFERENCE" or "ONLINE"
- `"BookName": string` containing one or multiple words in alphabet and digital number, length<60
- `"BookAuthor": string` containing one or multiple words in alphabet and digital number, length<20
- `"PageNumberArray":array(integer)` all elements are positive integers no more than 100000000 and separated by a semicolon, array length in [1, 200], no duplicated element.
- `"ConferenceName":string` containing one or multiple words in alphabet and digital number, length<60
- `"ConferencePlace":string` containing one or multiple words in alphabet
- `"WebsiteName":string` containing one or multiple words in alphabet and digital number, length<60
- `"WebHttpLink":string` string length<200
- `"Date":integer` in range [1,31]
- `"Month":integer` in range [1,12]
- `"Year":integer` in range [1, 2050]

We guarantee that if there are m ($m <= 20$) authors, each author[x] with $x \text{ in } [1, m]$ will appear at least once in the input. No duplicated string name will appear in the input. No special characters such as quote mark will appear in the content.

Please notice that the input order of the strings is not guaranteed, and the strings that do not appear in the input are regarded as null.

Output

The output is the string `{LANRAX}`:

`{LANRAX} = <@ {AUTHOR}; {PUBLISHER}; {TITLE}; {CITATION}; {TIME}.>`

No line break at the end of the output.

Example Input 1

```
13
"Author1FirstName":"vince"
"Author1FamilyName":"black"
"Author2FamilyName":"Lanran"
"City":"Shenzhen"
"Province":"Guangdong"
"Publisher":"VCN Center"
"Title":"How to set fire when you become a ghost"
"Type":"ONLINE"
"WebsiteName":"Lanran Club"
"WebHttpLink":"http://lanran.club/"
"Date":"9"
```

```
"Month":"11"
"Year":"2019"
```

Example Output 1

```
@ V. Black and Lanran; Shenzhen, Guangdong: VCN Center; "How to set fire when you become a ghost"; Accessed online from Lanran Club, link:http://lanran.club/; 9 Nov. 2019.
```

Example Input 2

```
15
"Author1FirstName":"vince"
"Author1FamilyName":"black"
"Author2FamilyName":"Lanran"
"Author3FamilyName":"Hong"
"Author4FamilyName":"Yee"
"Author5FamilyName":"Blue"
"Author6FirstName":"Stefan"
"Author6FamilyName":"Nafets"
"Publisher":"VCN Center"
"Title":"Data Structure in CS203"
"Type":"BOOK"
"PageNumberArray":"[1,2,5,3,7,9,12,13]"
"BookName":"Learn Data Structure Well in 1 Hour"
"Date":"9"
"Year":"2019"
```

Example Output 2

```
@ V. Black et al.; VCN Center; "Data Structure in CS203"; From Learn Data Structure Well in 1 Hour, page 1-3, 5, 7, 9 and 12-13; 2019.
```

Example Input 3

```
6
"Author1FirstName":"Bob"
"Author1FamilyName":"Huntington Whiteley"
"Title":"I start to believe no one"
"Date":"6"
"Month":"12"
"Year":"2019"
```

Example Output 3

```
@ B. Huntington Whiteley; ; "I start to believe no one"; ; 6 Dec. 2019.
```

Three is better than two

Constrain

`1s, 32M`

Description

Define the following sequence:

$$S_0 = s_0$$
$$S_n = S_{n-1}^2 \mod 2\,147\,483\,231, \forall n \geq 1$$

For a sequence in this form, Lanran has s_0 and s_{end}, which is the value that represents the end of the sequence. It is guaranteed s_{end} will appear, and the sequence ends at the first s_{end} with the length not exceeding $7\,100\,000$. $0 \leq s_0 < 2\,147\,483\,231$.

Now Lanran has three sequences in this form, he wants to find the longest common subsequence among these three sequences.

Please notice the memory limit!

Input

First line will be an integer T ($1 \leq T \leq 10$) represents the number of test cases.

For each test case, there will be a single line contains three s_0 and one common s_{end} separated by spaces, in the form of $s_0^1\ s_0^2\ s_0^3\ s_{end}$.

Output

For each test case, output one line contains one integer represents the start value of the the longest common sequence.

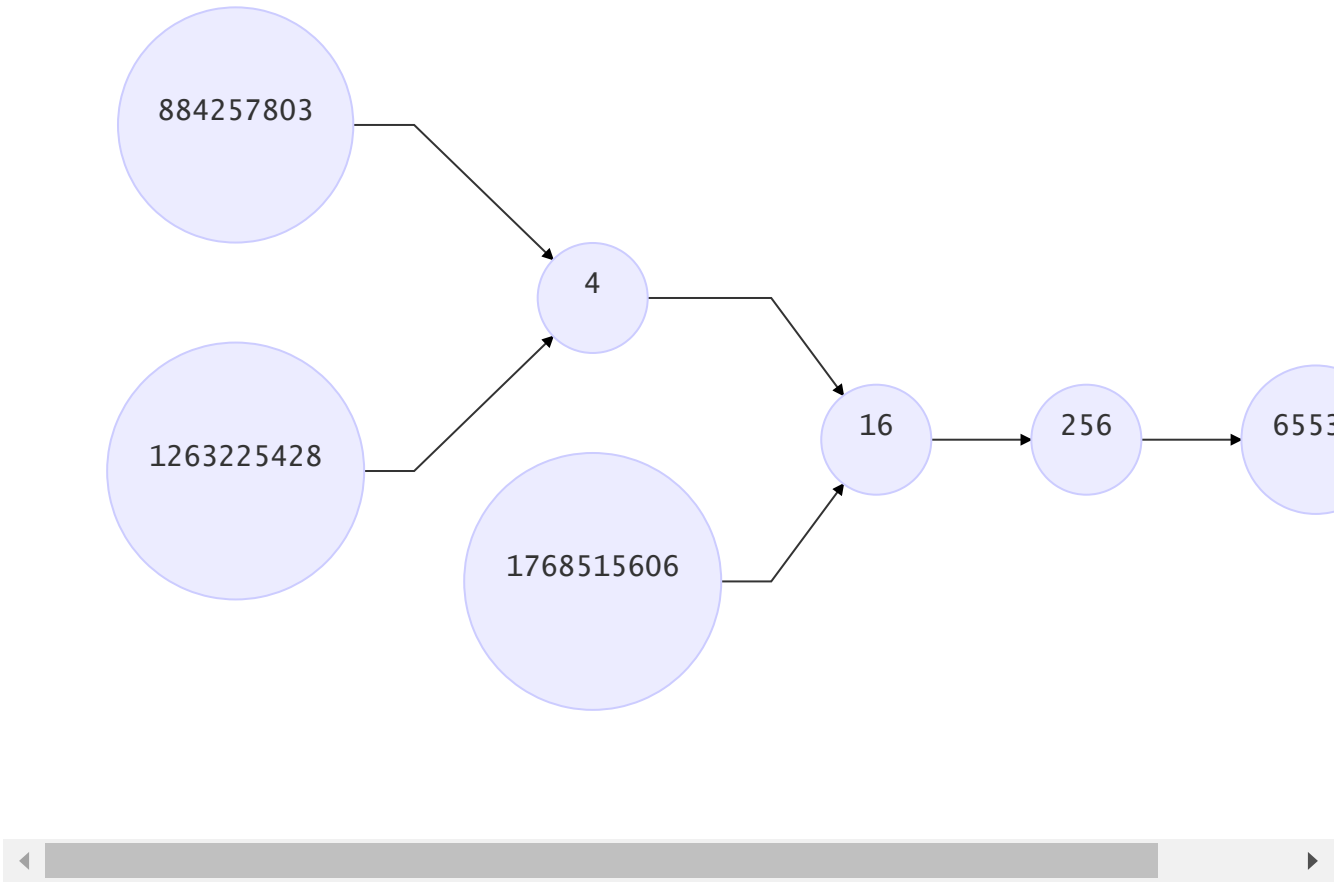
Sample input

```
2
884257803 1263225428 1768515606 65536
1 631612714 2147483230 1
```

Sample output

```
16
1
```

Hint


884257803
4
1263225428
16
256
1768515606

Graph

Constraint

`1s,256M`

Description

For a connected undirected graph $G =< V, E >$, we define a special triple (x, y, z) as an unstable triple if and only if $((x, y) \in E) \wedge ((y, z) \in E) \wedge ((x, z) \notin E) \wedge (x < z))$.

Given the number of nodes N and an integer K, please output the number of non-isomorphic graphs with exactly K unstable triples.

Input

The first line contain an integer T, $1 \leq T \leq 10$, indicating the number of test cases.

For each test case, there is one line containing two integers N, K, $1 \leq N \leq 100000, 2 \leq K \leq 7$.

Output

For each test case, output the number of non-isomorphic graphs with exactly K unstable triples. Because the answer could be extremely large, please output the answer modulo by 1000000007.

Sample input

```
3
3 1
4 3
4 4
```

Sample output

```
3
4
3
```

Begging for games

Constrain

`1s, 256M`

Description

Narnal is a beggar, nevertheless, his daily income is stable. Nowadays he wants to buy M games online. The good news is that all the games he wants are on special sales promotion, in which the price of each game will decrease K units day by day as long as the price can maintain a positive price value (if the price is 5, each day decrease 6, then the price should maintain at 5). The price of each game decreases on the morning of each day except the first day. However, Narnal adds his income to his wallet on the evening of each day including the first day.

Input

The first line will be four integers N, I, M, K, denote the original money Narnal had at the first day morning, the daily income of Narnal, the number of games and daily decrease units of each game respectively.

The second line will be M integers separated by spaces, denote the original price of each game, which would be in the range of $[1, 10^6]$.

$0 \leq N < 10^6, 1 \leq I < 10^6, 1 \leq M \leq 10^5, 0 \leq K \leq 10^6$.

Output

Print the minimum days Narnal can get all the games he wants and state at that time is whether morning or evening.

Sample input

```
1 1 3 1
2 1 1
```

Sample output

```
2 evening
```

Gifts

Constraint

`1s,256M`

Description

This year, Hong went to Shanghai and participated in International Collegiate Programming Contest (ICPC). After he has won the golden medal, he suddenly remembers his friends, Yee and Vince, so he decides to bring N gifts for them. Hong knows that the Vince's preference score for the i-th gift is a_i, and Yee's preference score for i-th gift is b_i. Each gift can only be given to one person, so Hong wishes to properly assign the gifts to maximize the summation of preference scores for every gifts. Moreover, Hong wants to give Vince at most A gifts and give Yee at most B gifts.

Input

The first line contains a integer T, $1 \leq T \leq 10$, indicating the number of test cases.

For each test cases there are three lines:

The first line contains three integers $N, A, B, 1 \leq A, B \leq N \leq 10^5$, indicating the number of gifts, the largest amount of gifts Vince could take, and the largest amount of gifts Yee could take.

The second line contains N integers, $a_1, a_2, \ldots, a_N$, indicating the Vince's preference score for each gift.

The third line contains N integers, $b_1, b_2, \ldots, b_N$, indicating the Yee's preference score for each gift.

$-10^9 \leq a_i, b_i \leq 10^9$

Output

For each test case, output the answer in a line, which is the maximum summation of the preference scores for the gifts received by Vince and Yee.

Sample input

```
1
2 1 1
1 3
2 4
```

Sample output

```
5
```

K-mod Tree

Constraint

`1s,256M`

Description

Once upon a time, there was a country ruled by the King of Modulo. The country had N cities, and all the cities were connected by $N-1$ roads between two cities, so that the graph of the country is actually a tree. For each road, there was a w to indicate the length. In this country, people could travel between the cities, but the travel fee was charged in a weird way, i.e., if a person travels from city u to city v, the travel fee is the distance between u and v modulo by a given integer K.

Now, here comes Q people, and each of them has a tuple (u, K) indicating the start city and the modulo. Please answer the largest travel fee by selecting an end city v for each tuple.

Input

The first line contains an integer N, $1 \leq N \leq 5000$, indicating the number of nodes on the tree.

On each line of the following $N-1$ lines, there are three integers u, v, w, $1 \leq u, v \leq N, 1 \leq w \leq 1000000$, indicating there is a undirected road with length w between u and v.

The $N+1$ line is an integer Q, indicating the the number of query tuples.

On each line of the following Q lines, there are two integers u, K, indicating the start city and modulo.

Output

For each query tuple, the largest travel fee by selecting an end city v should be outputted in a line.

Sample input

```
4
1 2 4
1 3 5
2 4 6
3
1 8
2 11
3 2
```

Sample output

```
5
9
1
```