Querying the Document

A document is represented as a collection paragraphs, a paragraph is represented as a collection of sentences, a sentence is represented as a collection of words and a word is represented as a collection of lower-case ([a-z]) and upper-case ([A-Z]) English characters.

You will convert a raw text document into its component paragraphs, sentences and words. To test your results, queries will ask you to return a specific paragraph, sentence or word as described below.

Alicia is studying the C programming language at the University of Dunkirk and she represents the words, sentences, paragraphs, and documents using pointers:

- A word is described by `char*`.
- A sentence is described by `char**`. The words in the sentence are separated by one space (" "). The last word does not end with a space(" ").
- A paragraph is described by `char***`. The sentences in the paragraph are separated by one period (".").
- A document is described by `char****`. The paragraphs in the document are separated by one newline("\n"). The last paragraph does not end with a newline.

For example:

Learning C is fun.
Learning pointers is more fun.It is good to have pointers.

- The only sentence in the first paragraph could be represented as:

```
char** first_sentence_in_first_paragraph = {"Learning", "C", "is", "fun"};
```

- The first paragraph itself could be represented as:

```
char*** first_paragraph = {{"Learning", "C", "is", "fun"}};
```

- The first sentence in the second paragraph could be represented as:

```
char** first_sentence_in_second_paragraph = {"Learning", "pointers", "is", "more", "fun"};
```

- The second sentence in the second paragraph could be represented as:

```
char** second_sentence_in_second_paragraph = {"It", "is", "good", "to", "have", "pointers"};
```

- The second paragraph could be represented as:

```
char*** second_paragraph = {{"Learning", "pointers", "is", "more", "fun"}, {"It", "is", "good", "to", "have", "pointers"}};
```

- Finally, the document could be represented as:

```
char**** document = {{{"Learning", "C", "is", "fun"}}, {{"Learning", "pointers", "is", "more", "fun"}, {"It", "is", "good", "to", "have", "pointers"}}};
```

Alicia has sent a document to her friend Teodora as a string of characters, i.e. represented by `char*` not `char****`. Help her convert the document to `char****` form by completing the following functions:

- `char**** get_document(char* text)` to return the document represented by `char****`.
- `char*** kth_paragraph(char**** document, int k)` to return the k^{th} paragraph.
- `char** kth_sentence_in_mth_paragraph(char****document, int k, int m)` to return the k^{th} sentence in the m^{th} paragraph.
- `char* kth_word_in_mth_sentence_of_nth_paragraph(char**** document, int k, int m, int n)` to return the k^{th} word in the m^{th} sentence of the n^{th} paragraph.

Input Format

The first line contains the integer $paragraph\_count$.
Each of the next $paragraph\_count$ lines contains a paragraph as a single string.
The next line contains the integer q, the number of queries.
Each of the next q lines contains a query in one of the following formats:

- 1 The first line contains $1\ \ k$:
  - The next line contains an integer x, the number of sentences in the k^{th} paragraph.
  - Each of the next x lines contains an integer $a[i]$, the number of words in the i^{th} sentence.
  - This query corresponds to calling the function **`kth_paragraph`**.
- 2 The first line contains $2\ \ k\ \ m$:
  - The next line contains an integer x, the number of words in the k^{th} sentence of the m^{th} paragraph.
  - This query corresponds to calling the function **`kth_sentence_in_mth_paragraph.`**
- 3 The first line contains $3\ \ k\ \ m\ \ \ n$:
  - This query corresponds to calling the function **`kth_word_in_mth_sentence_of_nth_paragraph.`**

Constraints

- The text which is passed to the **`get_document`** has words separated by a space (" "), sentences separated by a period (".") and paragraphs separated by a newline("\n").
- The last word in a sentence does not end with a space.
- The last paragraph does not end with a newline.
- The words contain only upper-case and lower-case English letters.
- $1\ \leq$ number of characters in the entire document $\leq\ 1000$
- $1\ \leq$ number of paragraphs in the entire document $\leq\ 5$

Output Format

Print the paragraph, sentence or the word corresponding to the query to check the logic of your code.

Sample Input 0

```
2
Learning C is fun.
Learning pointers is more fun.It is good to have pointers.
3
1 2
2
5
6
2 1 1
4
3 1 1 1
```

Sample Output 0

```
Learning pointers is more fun.It is good to have pointers.
Learning C is fun
Learning
```

Explanation 0

The first query corresponds to returning the second paragraph with 2 sentences of lengths 5 and 6 words.
The second query correspond to returning the first sentence of the first paragraph. It contains 4 words.
The third query corresponds to returning the first word of the first sentence of the first paragraph.