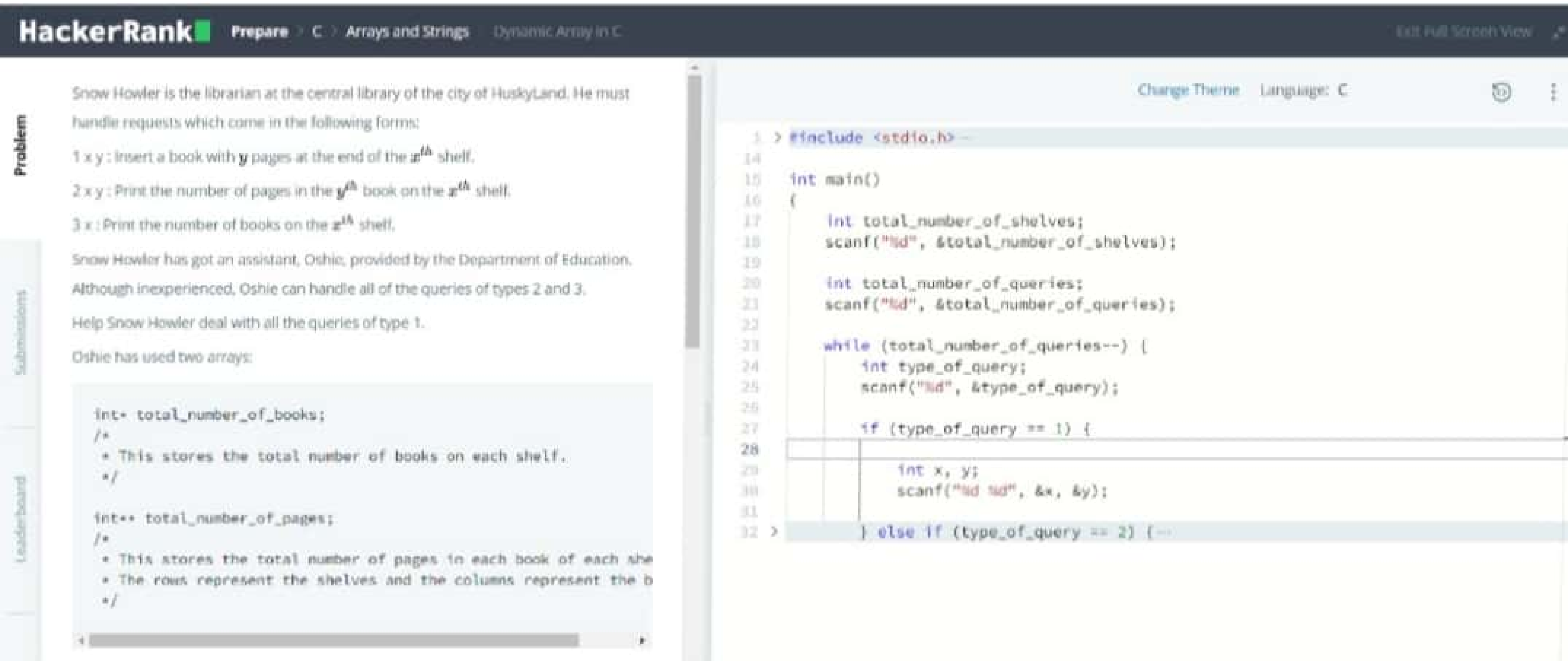

HackerRank
Prepare
C
Arrays and Strings
Dynamic Array in C
Exit Full Screen View
Problem
Submissions
Leaderboard
Change Theme
Language: C

Snow Howler is the librarian at the central library of the city of HuskyLand. He must handle requests which come in the following forms:

1 x y : Insert a book with **y** pages at the end of the xth shelf.

2 x y : Print the number of pages in the yth book on the xth shelf.

3 x : Print the number of books on the xth shelf.

Snow Howler has got an assistant, Oshie, provided by the Department of Education. Although inexperienced, Oshie can handle all of the queries of types 2 and 3.

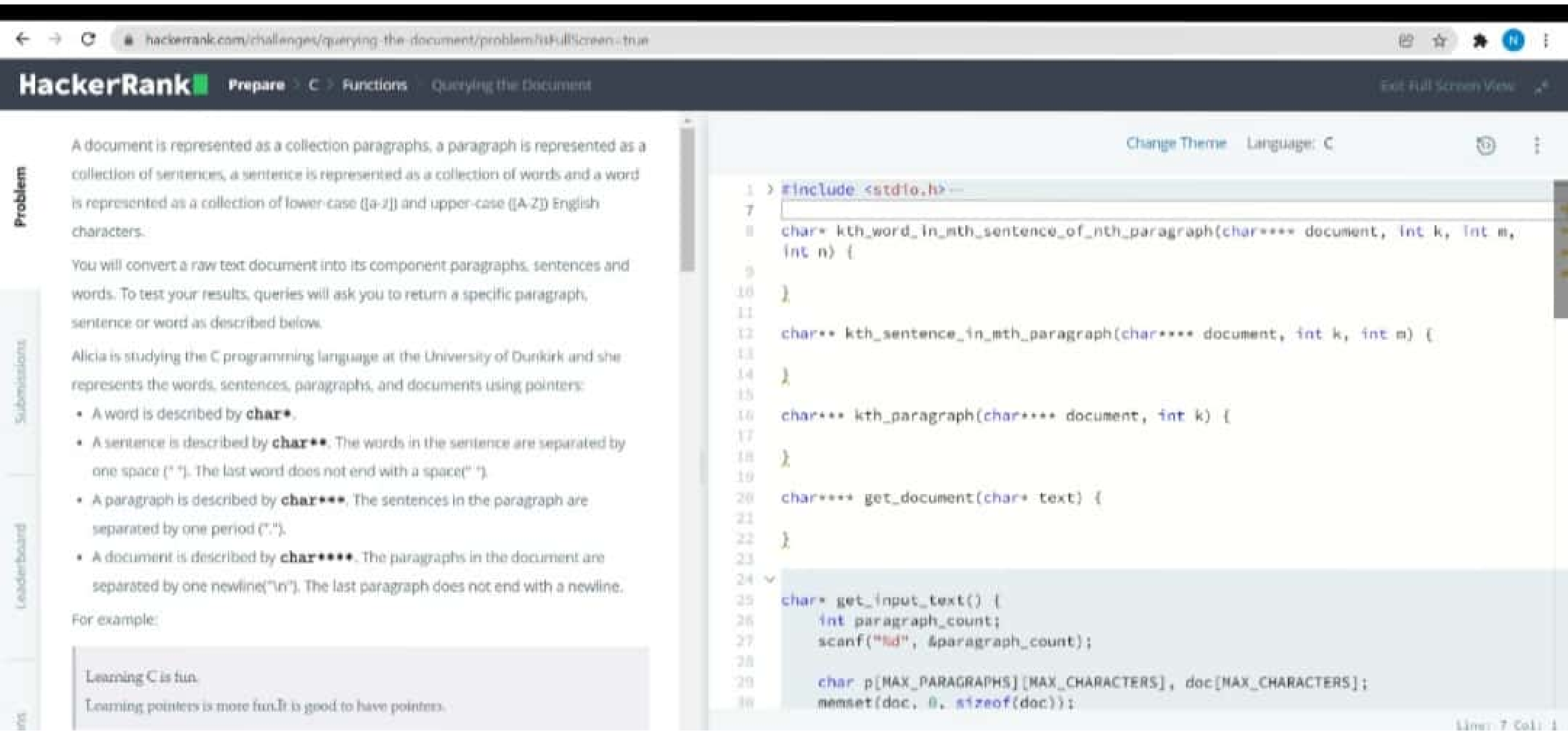
Help Snow Howler deal with all the queries of type 1.

Oshie has used two arrays:

```
int* total_number_of_books;
/*
 * This stores the total number of books on each shelf.
 */

int** total_number_of_pages;
/*
 * This stores the total number of pages in each book of each she
 * The rows represent the shelves and the columns represent the b
 */
```

```
> #include <stdio.h> …

int main()
{
    int total_number_of_shelves;
    scanf("%d", &total_number_of_shelves);

    int total_number_of_queries;
    scanf("%d", &total_number_of_queries);

    while (total_number_of_queries--) {
        int type_of_query;
        scanf("%d", &type_of_query);

        if (type_of_query == 1) {

            int x, y;
            scanf("%d %d", &x, &y);

> } else if (type_of_query == 2) { …
```

Scanned with CamScanner


hackerrank.com/challenges/querying-the-document/problem?isFullScreen=true
HackerRank
Prepare
C
Functions
Querying the Document
Exit Full Screen View
Problem
Submissions
Leaderboard
Change Theme
Language: C
Line: 7 Col: 1

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.

```c
#include <stdio.h>

char* kth_word_in_mth_sentence_of_nth_paragraph(char**** document, int k, int m,
int n) {

}

char** kth_sentence_in_mth_paragraph(char**** document, int k, int m) {

}

char*** kth_paragraph(char**** document, int k) {

}

char**** get_document(char* text) {

}


char* get_input_text() {
    int paragraph_count;
    scanf("%d", &paragraph_count);

    char p[MAX_PARAGRAPHS][MAX_CHARACTERS], doc[MAX_CHARACTERS];
    memset(doc, 0, sizeof(doc));
```

Scanned with CamScanner