Programming Challenge

Welcome to the DNAnexus programming challenge! This challenge is focused on your general programming ability, and is intended to be completed in two hours or less (though you may take more time as needed).

If you find anything confusing, just try to solve the problem as best you can. There isn't necessarily a single "right" answer - we are interested in seeing how you approach these types of problems, as well as your ability to write clean, correct code.

You may write the code in the language of your choice (please, nothing too exotic!), but it should be running code, not pseudocode. You're free to use standard library functions of the language you're using, but don't use a function that directly solves the problem (use your judgment). Please send us your original source code, preferably with a Readme file or comments explaining what it's doing.

Although it's not necessary, if you also created any test stubs and test data, feel free to send that as well.

If anything about the challenge is unclear, or if you have any feedback, please let us know! Thank you for your time and interest in DNAnexus, and good luck!

Problem 1: Random line from a file

You are given a very, very large plain text file where each line contains a plain text string. The file has at most *1 billion lines*; lines may have different lengths, but each line has at most *1000 characters*. Your goal is to write a program that will print an arbitrary line from the file. Your program will be run many times (although you don't know exactly how many times it will be run in advance), and you don't know in advance which lines might be selected. Thus, your solution should be optimized to minimize the runtime for each additional execution. The first execution of the program may take longer than subsequent runs, and you may use additional disk storage to improve performance.
Your program should take two command-line arguments: the *path* of the input file from which to print lines, and the *index* of the line you want to print. Your program should write the line to standard output.

Sample input/output:

input_file.txt:

```
apple
banana
orange
lemon
```

```
$ random_line input_file.txt 3
Writing index to input_file.txt.idx... done.
lemon
$ random_line input_file.txt 2
orange
$ random_line input_file.txt 0
apple
```

Problem 2: Search

You are given an array **A** of **n** integers. The elements of **A** are sorted in ascending order, and are not necessarily unique. You are also given a target integer **x**. Implement a search algorithm which finds a location j in the array such that all elements in the range **A[0], ..., A[j - 1]** are strictly less than **x**, and all elements in the range **A[j], ..., A[n - 1]** are greater than or equal to **x**. (Note that **x** itself need not appear in **A** for these conditions to be satisfied.) If no suitable location is found, return **-1**. Your solution should have runtime **O(log n)**.