Problem:

For this assignment, you will create a multithreaded Shannon-Fano-Elias encoder (https://en.wikipedia.org/wiki/Shannon−Fano−Elias_coding).

To determine the codes for the symbols using the Shannon-Fano-Elias encoder, you must execute the following steps:

1. Arrange the symbols according to decreasing probabilities.
2. Calculate cumulative probability.
3. Calculate modified cumulative distribution function.
4. Find the length of the code.
5. Generate the code word by finding the binary of Fbar(x) with respect to length l(x).

Given the symbols with their probabilities (sorted in decreasing order based on the probability value), you need to implement the Shannon-Fano-Elias encoding algorithm based on the following steps:

- ○ Read the input from STDIN (the Moodle server will implement input redirection to send the information from a file to STDIN). The format of the input file is as follows:

- A string representing the symbols. A single character represents each symbol, and a single white space separates the symbols.

- A list of double values representing the probabilities of the symbols

The symbols from the input are sorted (in decreasing order) based on their probabilities.

Given the previous format, the following file represents a valid input file:

```
A B C D E
 0.3 0.25 0.2 0.15 0.1
```

- ○ Create n POSIX threads (where n is the number of symbols to encode). Each child thread executes the following tasks:

- Receives the probabilities of the symbols from the main thread.

- Implements the Shannon-Fano-Elias encoding algorithm to determine the code for the assigned symbol.

- Stores the code on a memory location accessible by the main thread.

- Print the Shannon-Fano-Elias codes for the symbols from the input file. Given the previous input, the expected output is:

```
SHANNON-FANO-ELIAS Codes:

Symbol A, Code: 001
Symbol B, Code: 011
Symbol C, Code: 1010
Symbol D, Code: 1101
Symbol E, Code: 11110
```

NOTES:

- Not using multiple POSIX threads to implement your solution will translate into a penalty of 100%.
- You must use the output statement format based on the example above.
- You can safely assume that the input will always be valid.