Practice Problem on Array of Structures. **Try to solve this problem within a time limit of 1.5 hours (90 minutes).**

BUY-AND-SELL (contributed by F. R. Salvador)

1. INTRODUCTION

In this problem, we simulate how we can represent and process data for online buying and selling which has become more prominent due to the COVID-19 pandemic.

Consider two arrays of structures:

1. **SELLERS[]** – with each element storing information about a **seller** in a Buy-and-Sell platform.
2. **PRODUCTS[]** – with each element storing information about a **product** being sold in a Buy-and-Sell platform.

Seller Structure Information

Structure Member Name	***Type of Data***	***Purpose (or meaning) and example value***
ID	Integer	**Unique number** identifying the seller; for example: 1234
name	A structure made up two members namely: **first** and **last** both of which are at most 30 characters long *excluding* the null byte (all in capital letters)	Identifies the seller by name; for example: "MARIA" (the First name) and "CRUZ" (the Last name)
rating	Single precision float value	Value ranging from 0.0 to 5.0 indicating how reliable or trustworthy the seller is. More trustworthy sellers have higher ratings.

Product Structure Information

Structure Member Name	***Type of Data***	***Purpose (or meaning) and example value***
ID	Integer	Unique number identifying the seller; for example: 1234
product_type	Made up of at most 30 characters excluding the null byte (all in capital letters)	Identifies the type of product being sold, for example: "PEN"
price	Single precision floating point	Unit price; for example: 15.25

Some Notes:

a. As indicated above, each seller **ID** is unique, i.e., no two sellers have the same ID.
b. There maybe two or more sellers who are selling the same type of product. For example, seller **ID** 1234 is selling **product_type** "PEN", and another seller with **ID** 8765 is selling also the same **product_type** "PEN".
c. A seller may sell more than one type of product. For example, seller **ID** 1234 is selling both "PEN", and "PAPER". This means that there are two structures in the **PRODUCTS[]** array with an **ID** member equal to 1234, but with differing values for the **product_type** member.

2. YOUR TASKS:

1. Edit the accompanying header file MYHEADER-LASTNAME.h to accomplish the ff:
   a. Declare a data type for the seller structure. Use **sellerTag** as tag name. The names of each structure element should be exactly the same as those given in the table above (names are in blue font and yellow background).
   b. Declare a data type for the product structure. Use **productTag** as tag name. The names of each structure element should be exactly the same as those in the table above (names are in blue font and yellow background).
2. Edit the accompanying C source file BUYSELL-LASTNAME.c to accomplish the ff:
   a. Implement **InputSellers(SELLERS, n_sellers)** that will initialize the elements of the SELLERS array via scanf().
   b. Implement **InputProducts(PRODUCTS, n_products)** that will initialize the elements of the PRODUCT array via scanf().
   c. Implement **Linear_Search(SELLERS, n_sellers, search_key)** which should return the index where the seller structure with an ID member matching the search key parameter was found. If the key is not found, the function should return -1.
   d. Implement **Recommend(SELLERS, PRODUCTS, n_sellers, n_products, item).** The function should search for the highest rated seller, who is selling the item (i.e., the search string) at the lowest price. The following rules must be obeyed:

      Rule 1. Only sellers with a rating of at least 3.0 are qualified to be considered for recommendation.

      Rule 2. Tie-Breakers:

      Q1: What if there are two or more sellers selling the same product for the same price?
      A1: It is the seller with the highest Rating who is recommended. We value more Trustworthy people!

      Q2: What if the situation above happens, and the sellers have the same rating?
      A2: It is the seller with the smallest ID number (the lower the number, the more senior is the seller in the community) who is recommended. We respect people who are more senior than us!

      The function prints the name of the recommended seller and the price. In case the item is not found, the function prints "UNAVAILABLE"

The file **`INPUT.txt`** contains sample input data. Use input redirection technique to minimize the time needed for keyboard input of data. The file **`EXPECTED.txt`** contains the expected result following the sequence of function calls inside **`main()`**.

終