Computer Architectures – Exam test

A retailer must supply his shop by purchasing the goods from a wholesaler. The products sold by the wholesaler are listed in a `Price_list` table of `N` lines (`N` is a constant defined with `EQU`). Each line contains:

- a hexadecimal code uniquely identifying the product, expressed in 1 word (4 bytes)
- the price of one unit of the product, expressed on 1 word.

`Price_list` lines are sorted according to the growing identification code, but the identification codes are not necessarily consecutive (out-of-production products are removed from the wholesaler's catalogue).
The seller listed the products to be purchased in an `Item_list` table of `M` lines (`M` is a constant defined with `EQU`). Each line contains:

- the product identification code, expressed on 1 word
- the quantity of the product to be purchased, expressed on 1 word.

`Item_list` lines are not sorted.

You are asked to create a project with keil, replace the contents of the startup_LPC17xx.s file with the one in the **template directory** and **add the other files in the directory. Finally, write debugged** and **working subroutines** that meet the following 3 specifications.
Note 1: You should not change the code calling the subroutines. You are only required to implement subroutines.
Note 2: Specifications must be developed in order. You can only switch to Specification 2 after verifying that the solution to Specification 1 is working correctly. Same for Specification 3.

Specification 1 (8 points). Write a subroutine `sequentialSearch` that calculates the expense incurred by the shopkeeper to purchase the products from the wholesaler.
The subroutine receives the following parameters in the order indicated:

- address of the array `Price_list`
- address of array `Item_list`

Through a linear search (with a double while loop), the `sequentialSearch` subroutine calculates the total expense and returns it as a return value.
The subroutine shall be in accordance with the ARM Architecture Procedure Call Standard (AAPCS) for switching input/output parameters and saving registers.
Assume that every identification code in `Item_list` also exists in `Price_list`.

Specification 2 (6 points). Write a `binarySearch` subroutine (according to AAPCS standard) that calculates the expense incurred by the shopkeeper to purchase the products from the wholesaler.
The subroutine receives the following parameters in the order indicated:

- address of the array `Price_list`
- address of array `Item_list`

The `binarySearch` subroutine uses the `Price_list` sorting based on the identification codes to find the products through a binary search. Refer to the pseudocode of binary search:

```
first = 0;
last = N - 1;
index = 0;
while (1)
{
     middle = (first + last) / 2;
     if (key == table[middle])
```

```
      {
            index = middle;      /* element found */
            break;
      }
      else
            if (key < table[middle])
                  last = middle - 1;
            else
                  first = middle + 1;
}
```

Assume that every identification code in `Item_list` also exists in `Price_list`.

Specification 3 (5 points). Write appropriate instructions / functions in C to implement the following features:

- at the key KEY1: call the subroutine `sequentialSearch` passing as parameters the vectors `ItemList` and `pricelist` defined in the file sample.c
- at the key KEY2: call the subroutine `binarySearch` passing as parameters the vectors `ItemList` and `pricelist` defined in the file sample.c

In both cases, if the shopkeeper's expense is greater than the `MAX` constant defined in the file sample.c, then you must emit an acoustic warning signal lasting 2 seconds (volume and note can be chosen freely).