**ANKARA UNIVERSITY**

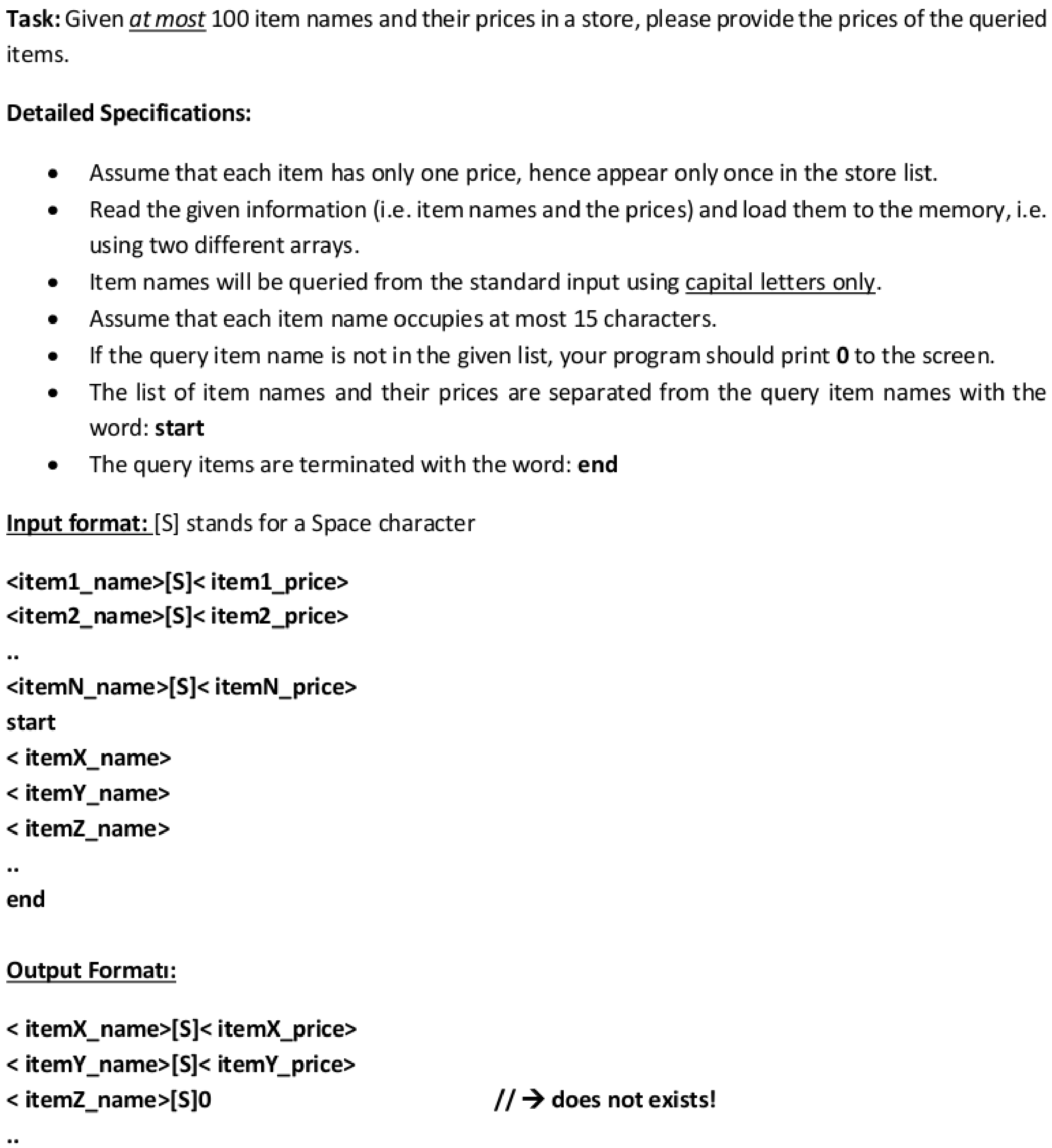
**COMPUTER ENGINEERING DEPARTMENT**

**Computer Programming II**

**Spring 2022-23**

Lab6 Quiz
Assist. Prof. Dr. İrem ÜLKÜ

Date: 12/05/2023
Duration : 90 mins.



**Task:** Given *at most* 100 item names and their prices in a store, please provide the prices of the queried items.

**Detailed Specifications:**

- Assume that each item has only one price, hence appear only once in the store list.
- Read the given information (i.e. item names and the prices) and load them to the memory, i.e. using two different arrays.
- Item names will be queried from the standard input using capital letters only.
- Assume that each item name occupies at most 15 characters.
- If the query item name is not in the given list, your program should print **0** to the screen.
- The list of item names and their prices are separated from the query item names with the word: **start**
- The query items are terminated with the word: **end**

**Input format:** [S] stands for a Space character

**<item1_name>[S]< item1_price>**
**<item2_name>[S]< item2_price>**
**..**
**<itemN_name>[S]< itemN_price>**
**start**
**< itemX_name>**
**< itemY_name>**
**< itemZ_name>**
**..**
**end**

**Output Formatı:**

**< itemX_name>[S]< itemX_price>**
**< itemY_name>[S]< itemY_price>**
**< itemZ_name>[S]0** **// → does not exists!**
**..**

**Hints:**

1. You can include <string.h> file and use related string functions if you need them.
2. You can use [100][15] two dimensional character array to store the item names.
3. You can use [100] one dimensional integer array to store prices.
4. When you load the item names to the character array, you should load the corresponding prices to the integer array with the same row index.

**Testing:**

We provided you a sample I/O file pair. Test your programs with these files while developing your program. Moreover, create your own test files according to the specifications to test more.

As usual, use *input redirection* mechanism of your operating system to test your programs. For example, if your executable is called as **Lab8**, redirect the input.txt file to standard input using **<** operator and redirect your outputs to a file using **>** operator such as:

```
> ./Lab8<input.txt>output.txt
```

This kind of execution enables your programs to read inputs from a file without writing any file related functions (e.g. fopen(), fscanf() etc.). In other words, the getchar() or scanf() functions in your code reads data from the redirected files instead of the std. input in this way (e.g. keyboard).

**Submission:**

Before submission, make sure that your source file is named as **StudentNumber.c**

Good luck ☺