

Specification for program “Mi carrito”

Authors: Group 7

NAME:

Mi carrito - management of grocery shopping list.

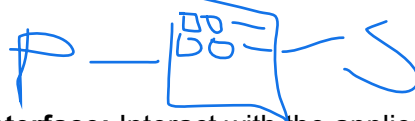
USAGE:

micarrito <input-file>

DESCRIPTION:

micarrito reads commands from an input file. The command-line application allows users to create and manage a shopping list efficiently. By entering commands, you can save products you want to purchase, along with the urgency level for each item, quantity, brand, and prices across different stores. Other commands include, show shopping list, delete products from the shopping list, change the urgency level or the quantity of the products, in addition to locating the store with the lowest total price.


Key Features:





- **Command-Line Interface:** Interact with the application entirely through the command line for quick and easy access.
- **Add Products:** Save items to your shopping list by specifying the product name.
- **Set Urgency Levels:** Assign an urgency level to each product (e.g., low, medium, high) to indicate how soon you need to buy it.
- **Manage Your List:** View, edit, or remove items from your shopping list as needed.
- **Prioritized Shopping:** The application organizes your list based on urgency, helping you prioritize your purchases.
- **Data Storage:** All products and their urgency levels are saved locally for future reference.
- **Store Management:** Add or remove stores, along with product prices in each store.
- **Find Cheapest Store:** Identify the store with the lowest total cost for the shopping list.

OPTIONS:

The program must process input files containing specific commands, each written on a new line, and the format is case-sensitive, meaning lower and uppercase letters must be carefully distinguished. Commands are limited to a maximum line length of 1024 characters, with spaces only allowed between commands and arguments, but not within the arguments themselves. Each command should be followed by one or more arguments, with a single space separating the command from its arguments. The program must handle different numbers of arguments per command while strictly adhering to the requirement that commands are written on separate lines and follow the correct case-sensitive format.

- **show <CartName>:** shows all the products that are in the shopping list. 



- **showUrg <UrgencyLevel>**: shows all the products that are in the cart based on the urgency level specified.
- **addProd <ProductName> <UrgencyLevel> <Quantity> <Price>**: allows users to add products to the shopping list, including the level of urgency and the quantity, and the price of it.
- **remvProd <ProductName>**: allows users to remove the specified product from the list.
- **editUrg <ProductName> <UrgencyLevel>**: allows the user to change the level of urgency of a certain product.
- **editQuantity <ProductName> <Quantity>**: allows the user to change the quantity of a certain product.
-  ➤ **editPrice <ProductName> <StoreName> <Price>**: it will update the price of the given product in the specified location.
- **addStore <StoreName>**: will include the new store to the list of those available.
- **remvStore <StoreName>**: it will remove the given store, along with the prices of the associated products.
-  ➤ **findCheapestStore**: will display the name of the store that gives the lowest total price of the shopping list, along with the total price.
- **addBrand <ProductName> <BrandName>**: allows to add or update the brand of an existing product.

INPUT DATA

None

LIMITATIONS

The commands read from the input file must be written in the exact format as specified, or the program will not function correctly.

If a product is already in the list and you use the add command with the same product name, it will update its urgency level to the new one provided and **update the quantity**. 

Urgency levels must be one of the accepted values: LOW, MEDIUM, HIGH. Any other values will result in an **error**.

If a **product** is not in the list, attempting to remove or update it will result in an **error message**.

Stores and prices must be added separately for each product if multiple stores are used.

When adding a product, the BrandName must be provided. This command can be used to **update the brand of an existing product**.

The program will output specific error or warning messages based on the encountered issue. For instance, if a command is not recognized, the system will display an error message stating: "Error: Invalid command '<CommandName>'." Similarly, if the number of arguments provided for a command is incorrect, it will show an error message such as: "Error: Incorrect number of arguments for command '<CommandName>'."