Final Project

Inventory Management

Due: April 9th, 11pm

(No submissions accepted after this time)

RestEasy is a company which stores products in different warehouses. Since they've increased their range of products, they have been having trouble keeping track of all of their items. You are being tasked with creating an inventory management system to keep track of the company's stock. Every item they store has 5 attributes associated with it:

1. Item Number – a unique product identifier (e.g. 1000000)
2. Quanity – how many of the item are in stock (e.g. 10)
3. Item Name – the item name that is known by customers (e.g. BrandName 3mm Screws)
4. Item Location – which warehouse the item can be found in (e.g. Warehouse A)
5. Item Description – Additional information about the item (e.g. Requires a Philips screwdriver to use)

1 Requirements

Your program must include the following functionality:

- Maintains a list of items, sorted by Item Number
- Ability to add a new product to inventory; the inventory item list must remain sorted when an item is added
- Ability to delete an item from inventory with a confirmation button; you can use a dialog box or a second button (If you use a second button, keep it blank and disabled until the delete button is clicked, then change the second button's text to say "Confirm".)
- Ability to search for an item using the Item Number attribute as the search key; you must use the binary search algorithm to get full credit for this part
- Ability to display all attributes associated with an item
- Ability to update the Quantity, Item Name, Item Location, or Item Description of a specified item
- Ability to load and save inventory data using a text file

Your GUI will need, at minimum, the following:

- A text field for each of Item Number, Quantity, Item Name, Item Location, and Item Description
- A label for each entry field to describe its purpose
- A 'New' button that will insert create a new inventory record from the entry field data and add it to the inventory
- A 'Delete' button that will remove the displayed item from the inventory
- A 'Search' button that will take the value in the Item Number text field and fill in the GUI's remaining text fields with the found data
- An 'Update' button that will search for a given Item Number and alter the stored data with the new data provided in the GUI text fields

- A 'Load Data' button that will load inventory data from a formatted text file into the program; you must use GUI file selection for this part
- A 'Save Data' button that will write the current inventory data into a text file; you must use GUI file selection for this part

You should add any additional GUI elements that you need to make a nice-looking interface. **You must use `tkinter` and/or `tk/ttk` to build your GUI.** We recommend using the `grid()` option for widget placement, but you may use any method that you prefer that are available in `tkinter` or `tk/ttk`.

Exception Handling: [worth about 8%]
Your program should perform exception handling. For example, invalid Item Number values, including an blank entry, should be detected using a try/catch statement. Negative numbers or other input errors should not cause your program to crash. You should also not allow invalid data to be stored in the inventory, nor written to files. If invalid input is given, the program should inform the user that the input was invalid. This list is not exhaustive, so come up with as many situations as you can that requires exception handling.
For full marks, your program should also state what was wrong with the input. How you do that is up to you. One example might be using a label to display error messages.

Naming Conventions:
Name variables and functions according to the actions they perform (e.g. If your function prints an introduction, call it `printIntro` instead of `functionA`; `iName` or `itemName` or `nameOfItem` are all good variable names, but `variable3` is not).

Comments: [worth about 2%]
Document what functions do and use broad comments to describe any significant sections in your code such as code to display related GUI elements. You can add in a line of hashtags # to separate out chunks of code which do different things.

2 Testing Your Project's Load Functionality

I recommend creating a test file (or possibly several) to use for testing, but do not submit them as part of your project files. The data will be loaded from a text file, which should be set up like this:

```
1111111,170,wrenches,Warehouse B,steel wrenches
3333333,133,screws,Warehouse A,standard screws
2222222,21,screwdrivers,Warehouse C,Large philips screwdrivers
```

When the user clicks the 'Load' button, your program must ask the user to navigate to the text file they want to use using a GUI file selection window. It should then load all data into system memory. You can assume that the inventory will easily fit in working memory.
To be able to efficiently search for items, we must ensure that the inventory is sorted at all times. Since the text file we are reading from might not be sorted, you must sort the inventory records by Item Number before any other operations can be called.

3 Suggestions and Hints

Initially, this project looks like a very large task. Breaking it down into different tasks will make it look more manageable.

- Decide what you want your GUI to look like. This will help you with widget organization and placement. I suggest drawing something on paper to help you visualize the final result.

- You should begin by placing all the elements that you need on the GUI and building an `Item` class to store inventory records. These two things can be done independently since they will not rely on each other until you program their interaction. Make sure you have a running program at this stage. Programs without these early stages will receive very few marks.
- Create the 'Load' button functionality. Again, break it down into steps:
  - Add code so that when the user clicks the button, it asks for a file to be selected.
  - Create a function to load data from a file and pass it the file name from the file selection GUI widget. Your loading functionality should include the creation of an `Item` object for each line in the file. If you are having trouble creating `Item` objects from a text file, try testing your program on a file with one line first.
- The 'Save' button functionality should be similar to 'Load', so implement this next except this time it should write the contents of `Item` object array to a text file. The user should be able to select a file using GUI file selection. Now you can test out your other functions as you develop them.
- Chapter 13 in the textbook presents two sorting algorithms. For full marks, implement the mergesort algorithm. If you choose to use the much simpler selection sort, you can still receive most of the marks for sorting.
- If the user clicks on the 'Update' button, your program must change all the attributes of the item entry matching the given Item Number to the information written in the Entry fields. In practice, users should search for an Item Number and modify the field they wish to change before clicking 'Update'. This ensures that users know what they are changing and prevents data from being unintentionally overwritten. At minimum, you should make sure that every text field has a value in it before making the 'Update' button available to the user.
- Once you have completed and tested all of the required project components, you should customize your GUI by using different fonts, colours, and sizes. If you are feeling adventurous, you can move your `Item` class into a separate file and import it into your inventory program.

4 Submission Notes

Call your submission `lastname_Project.py`
If your project does not run, you will get a zero. We need to see the runtime behaviour as well as the underlying code. Only include the components that you have completed or that do not add errors. You may comment out component attempts that do not work to get some credit for them. You should make it clear what these sections are for. If we do not know what you were trying to do, we cannot give you marks for it.