# Inventory Data Structure

## Exercise Questions

There is also a level indicated for each question. Level 1 is the easiest and level 5 hardest. The exercises can be tried out as an assignment of data structure course.

**Question 01:**

There are new data sets given in the directory. Get all the options done for the new data set as supplied in the main.c. This task is easy. You just need to verify the functions on the new data set.

**Level: 1**

**Question 02:**

For the data set you wish to work on, add the following functionality: Write a function to update an existing value to new value in the table. User needs to mention the index where the update has to be made and also supply the new value. A confirmation has to be taken from the user before updating to new value.

**Level: 2**

**Question 03:**

For the data set you wish to work on, add the following functionality: Write a function to convert between the data types. You know that table houses integer data as well in the form of strings. Write a function to convert such strings to integer values. You need to devise a mechanism to only convert the valid fields. Example: the field 'car' should not be converted to integer type.

**Level: 2**

**Question 04:**

We all know that to add a new key the data is read from the file. No check is made if the file data supplied is in the proper format. Write a function to check if the supplied input file has the data written in the required valid format.

**Level: 2**

**Question 05:**

Design a query processor. User has to give the input in the following way:

'property condition value'

Where 'property' is a valid property from inventory table, 'condition' is conditional operator among <, >, ==, <=, >=, != and 'value' is an integer value.

Firstly parse the input for validity, then evaluate it and print the results to the user.

**Level: 3**

**Question 06:**

Incorporate the feature of adding a new property to the data structure. Remember that adding a new property has to add a value to all the existing keywords. Count has to be updated and re written in file. Suitable changes have to be made while adding a new key too.

**Level: 4**

**Question 07:**

Add the feature to calculate the hash value of inventory data. The hash value has to be used to check the consistency. After every operation user does, if there was an access made to inventory data, then the hash has to re-computed and verified if the data is altered from the table. Use a secure hashing algorithm.

**Level: 4**

**Question 08:**

We have the inventory table as combination of linked lists rather than arrays. Essentially, the table is having header nodes which house the values of keys. Attached to every key is a linked list which carries the properties. Inventory table is a collection of linked lists and the entire keys together form a linked list as well. Implement the add_key() and print() for this table.

**Level: 5**

***Note:***

*It's a PH and VT effort*