**Question 1 (20%): A simple sales records management system (**[**Back to Beginning**](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/index.shtml)**)**

Western Wholesales Pty Ltd sells variety of products to the re-seller markets. It sells products in the following 6 categories: food, alcohol and beverage, apparel, furniture, household appliances, and computer equipment. Design a program to manage the sale records of the company and implement the program in Python programming language.

These sales records are stored in two CSV files, on for the customer details and the other for the records of sales to these customers.

The first file contains the customer records. Each customer record consists of following information: the customer id cust\_id, the customer's name, the customer's postcode, the customer's phone number. In each record, only the customer's ID and name are required. Other details are optional (empty). Note, the customer id is a 6-digit number starting from 100000.

The second file contains the sale records. Each record contains the details of one sale, including date, trans\_id, cust\_id, catetegory, and value. Note the transaction id (trans\_id) is a 9-digit number starting from 100000000.

Design a program and implement it in Python programming language that would load the customer records and sales records from two CSV files and then store the customer and sales details in a single data structure in the computer memory.

The two file names/paths can be optionally supplied as two command line arguments, with customer records file comes first, followed by the sales records file.

The program repeatedly displays a menu to allow the user to perform the following operations:

* Load customer and sales records from two CSV files and store them in a single data structure. If there is already a data structure containing customer and sales records in the memory (either via the two CSV files from the command line or from a previous load customer and sales records menu option), the new customer and sales records from two new files would replace the existing customer records and the sales records in the computer memory. The program should prompt the user to provide the names/paths of the two new CSV files.
* Save all customer records from the memory to a CSV file. Your program should prompt the user to provide a file path. If the file does not exist, create a new file. If the file does exist, warn the user that the content of the file would be lost and give the user the option of either changing the file name or cancel the operation. If the user decides to overwrite the file, save the customer records to that CSV file. The saved customer records file should have the same structure as that of the sample customer records file [customers.csv](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/customers.csv). If there are no customer and sales records in the memory, inform the user and return to the main menu.
* Save all sales records from the memory to a CSV file. Your program should prompt the user to enter a file path. If the file does not exist, create a new file. If the file does exist, warn the user that the content of the file would be lost and give the user the option of either changing the file name or cancel the operation. If the user decides to overwrite the file, save the sales records to that CSV file. The saved sales records file should have the same structure as that of the sample sales records file [salles.csv](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/sales.csv). If there are no customer and sales records in the memory, inform the user and return to the main menu.
* Quit the program.

Restrictions:

* in this question, do not use ndarrays to store the customer records or sales records.

Recommendations:

* You may use Python's builtin module csv to handle CSV files.

For your convenience, I have created two test files [customers.csv](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/customers.csv) and [sales.csv](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/sales.csv). The customer records file contains the details of 200 customers while the sales records file contains 1000 sales records. The information inside the files are completely fititious and were randomly generated.

You are required to provide test evidence in your assignment documentation that your program was tested successfully using the above test files

You should test your program using multiple test data sets. You may create a few small test data sets of your own, perhaps containing a dozen customers and several dozens of sales records, and use these small test data sets to test your program first. Once your program works correctly with these small test data sets, you should then test your program using the above large test data set.

When creating your own test files, use the same column names as in [customers.csv](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/customers.csv) and [sales.csv](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/sales.csv).

**Question 2 (40%): Search and manipulation of customer and sales records (**[**Back to Beginning**](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/index.shtml)**)**

Revised your solution to [Question 1](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/index.shtml) by adding new features in the form of additional menu options:

* Add the details of a new customer to the data structure containing customer and salles records. The customer details include the cust\_id, the customer's name, the customer's postcode, the customer's phone number. When the user enters the details, only the customer's name is required, other details are optional (leaves it empty). Note, the customer id must be unique and be automatically generated, which must start from 100000 with 6 digits. For convenience to the user, the auto-generated customer id should be displayed after the customer record is successfully added.
* Add a new sales record for an existing customer. In this case, the user needs to provide the customer id. If the customer id exists, the user can then enter the details of a new sales record, ie, date, trans\_id, cust\_id, catetegory, and value. Note for each sales record, there must be a unque, auto-generated, transaction id for that sales record which starts from 100000000 with 9 digits. If the user provided customer id does not exist, inform the user that there is no such customer and ask the user to provide a correct customer number or return to the main menu.
* Search customers using a single search string. The search string is compared to the customer's id, name, postcode and phone number. Then the program will display the details of all matching customers. The search must be case-insensitive and must allow partial matches, for example, the search string "john" would match "John Smith" as well as "Elton Johns"
* Search sales records using a single search string. The search string is compared to customer id, date, category and value. Then the program will display the details of all matching sales records. Note: just like searching for customers, search for sales records must allow partial matches.
* Display all sales records from a customer using his/her customer id. When the customer id is given, the program should display all sales records due to that customer.
* Delete a sales record with a given transaction id.
* Delete a customer with a given customer id together with all sales records due to that customer.

Restrictions:

* your design must be modular with at least two modules (in addition to the main module) in your implementation, each provides some closely related functionality.
* in this question, do not use ndarrays to store the customer records or transaction records.
* in this assignment, do not use any Data Base Management System or module.

**Question 3 (30%): Display sales performance graphically (**[**Back to Beginning**](https://www.it.murdoch.edu.au/~S900432D/krIuUnc93_hhxyePllj154_8xei/assignments/2024s1/index.shtml)**)**

In Question 1 and Question 2, you may have used one or two language builtin data types, such as list and/or dictionary, to store the customer and sales records. These builtin data types provide many useful methods which greatly simplified your programming. For small data size, this is an appropriate design choice. However, if the system is to be used to handle very large data sets, such as thousands of customers and tens of thousands (or even more) of transactions, these built-in data types may not be the most appropriate to use as they are less efficient compared to the ndarrays.

To improve the system efficiency, in this question, you will revise the program from Question 2 by storing the customer and sales records in a single ndarray and make the necessary adjustments to the rest of the program.

Furthermore, you will provide the following additional features for analysing the sales performance in the form of the following additional menu options:

* Display the monthly sales values and the number of sales using two line graphs in one axes. The line graph must have an appropriate title, the labels for X axis and Y axis, and a legend.
* For a given customer, display the monthly sales values and the number of sales due to the customer using two line graphs in one axes. The line graph must have an appropriate title, the labels for X axis and Y axis, and a legend.
* For a given postcode, display the monthly sales values and the number of sales due to the customers located in the postcode area using two line graphs in one axes. The line graph must have an appropriate title, the labels for X axis and Y axis, and a legend.

You should use Python modules numpy and matplotlib for this question.