Dublin Institute of Technology

Program with Persistent Data

Assignment

Lecturer Dr. Pierpaolo Dondio

You are required to implement a small database in C for an online store. The database contains the list of customers and the list of products that each customer bought.

The online store has already an old database consisting of two text files, one with information about customers (*customer.txt*) and one with information about sales (*sales.txt*).

For each customer, the following information is stored in the file *customer.txt*:

1. Customer ID (a unique numeric field)
2. Name
3. Surname
4. Age
5. Gender
6. Address

The customer ID is unique. The maximum size of the address field is 100 characters, while the maximum size of the name and surname is 40 characters. The above information are stored sequentially line by line in the file *customer.txt*, note that each of the fields is written on separate lines. Therefore the format of the file is:

4567
Abdul
Doyle
32
m
grafton street 15
4534
Svetlana
Reed
47
f
baker street 30
1345
Mike
Green
40
m
kevin street 23
……

Therefore there are 6 lines for each customer (id, name, surname, age, gender, address).

For each sale the store saves the following information:

1. Customer ID (the customer who bought the product)
2. Product Description
3. Unitary Price
4. Quantity

The product description is a string with maximum size 100 characters that describes the product (for instance “pen”, “laptop”, “lego city”….), the price is a variable double and the quantity is an integer. The Customer IDs are the ID of an existing customer in the file *customer.txt*. Again, the information are stored sequentially line by line in the text file *sales.txt*.

Therefore the format of the file is:

4567
cake
567.2
4
4567
lego
212.7
1
4511
lego
106.35
10
4567
iphone
319.05
9
….

Therefore there are 4 lines for each sale (customer id, product description, unitary price and quantity sold).

You are required to implement the following requirements:

1. Conversion to binary file
   You are required to read the two text files and create a binary version of them. This requires defining the proper record structure for customer and product sales and creating the file *customer.bin* and *sales.bin*
   At the end of the operation the files have to contain the same data as the original text files. You might decide to sort the binary files or not. The original text files are not sorted by customer ID.

2. Searching functionalities

   a. Show Customer Details by customer ID
   You are required to provide a search function by customer ID, that shows all the information about a customer contained in the *customer.bin* file (do not display any product bought by the customer)

   b. Show all the sales of a customer
   The search function gets a customer ID and displays all the products – with price and quantities – bought by that customer (ie the information in the *sales.bin* file)

3. *Add* functionalities
   a. Implement a functionality to add a new customer to the database. The customer ID is provided by the user. If the customer ID already exists, the customer is not added and an error message is displayed.

   b. Implement a functionality to add a new sale, by inserting the customer id, the product, the price and the quantity. If the customer ID does not exist (ie the customer does not exist in the file *customer.bin*) the function displays an error message.

4. Search by index
   Implement the above functionality using an index kept in memory (note that the index can be also stored in a file if you wish). You are required to build an index for the *customer.bin* file – using *customer id* as the key – and one index for the file *sales.bin* – using again *customer id* as the key, even if this time the key is not unique and it can have duplicates (note how this is not affecting your application, since the search function only requires to display all the sales related to a specific customer and not a specific sale).

Marking Scheme

Functionality	Marks
Conversion of the text files into binary files	10
Search customer functionality	8
Search sales by a specific customer functionality	8
Add Customer functionality	10
Add Sale functionality	12
Index creation (customers)	8
Index creation (sales)	8
Search / Add customer (using index)	12
Search /Add sale (using index)	14
Implementation of a text-menu for the application	10
TOTAL	**100**

Make sure your program compiles with no errors!