Exercise set 4

Structs, pointers, and dynamic memory

Introduction

In this homework, you will write a simple command-line program that will manage the products in the store. Your program will be able to add, remove, print, and update the products in the store.

Each product in the store will be described with several parameters such as barcode, name, price, expiration date, and so on. Each product will have a unique field, a barcode; no two products are allowed to have the same barcode.

Before each command, a program will print a generic menu that asks the user to choose the operation to perform. Possibilities include adding a product, removing a product, listing the expired products, listing all the products, and updating the product details.

Do not get scared from the length of the statement — it is meant to help you easily solve the problem.

You will get a skeleton program that will include all the messages that your program has to output. Try not to output any textual messages except for those already given to you, since it will affect the correctness of your program output.

Detailed description

Structs

In this assignment, we will use 3 structs.

1) Since every product has an expiry date, we shall build a `date` struct:

```
typedef struct date
{
	int year;
	int month;
	int day;
} date;
```

2) Every store has products, build a `product` struct:

```
typedef struct product
{
	char * product_name;
	char * product_category;
	char * barcode;
	int available;
	double price;
	date * expire_date;
} product;
```

3) The store struct `super_market`:

```
typedef struct super_market
{
	product ** product_list;
	int number_of_products;
} super_market;
```

Constants

Those constants should be implemented in your code:

```
#define MAX_NUM_PRODUCTS 20
#define MAX_PRODUCT_NAME_LENGTH 20
#define MAX_CATEGORY_LENGTH 10
#define BARCODE_LENGTH 12
```

`MAX_NUM_PRODUCTS` defines the max capacity of the store, meaning the upper limit of products that our store can contain.

`MAX_PRODUCT_NAME_LENGTH`, `MAX_CATEGORY_LENGTH`, `BARCODE_LENGTH` define the max input lengths of the appropriate field in product struct.

The interface

The program will show the user the following interface, where the desired action can be selected:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:
```

(USE 1 TAB IN ORDER TO CREATE THE SPACE BETWEEN EVERY OPTION).

Important note: the interface should be printed after every finished action except "EXIT SYSTEM".

Details of supported actions

1. Add product

When this option is chosen, the user is asked to enter all the relevant information about the product: product barcode, product name, product category, number of the available products (how many products of this type to add), product price, and the expiry date of the product.

Before asking for all the parameters, if the typed barcode already exists in the store the user will be asked to update the product quantity without new memory allocation.

If there is no more place to add products the program should print the following message: "Can't add more products, not enough space!".

Example:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:1234567890
Please enter product name:Milk
Please enter product category:Dairy
Please enter number of products to add:5
Please enter the price of the product:4.90
Please enter expiration date of the product[dd/mm/yy]:12/05/20
The product Milk -barcode:1234567890 ,added successfully
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:1234567890
This product already exist, please enter the number of products to
add: 3
Additional 3 products of Milk added
```

2. Remove product

In this case, the user will be asked to enter the barcode of the product that would be deleted. Deleting the product means freeing all the memory that this product used. If the product was not found, the appropriate message should be printed. Remember to update the number of products and free all the required fields.

Example:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:2
Please enter product barcode you want to delete:1234567891
Couldn't find the product barcode, try again...
Please enter product barcode you want to delete:1234567890
The product deleted successfully!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:2
The store is empty!
```

3. Check which products are expired

In this case, the manager wants to check which products have expired. The user is asked to enter a date, and the program should print all the items that have expired before this date. In other words, it should print all the products with an expiration date strictly before the entered date.

Example:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:0123456789
Please enter product name:Milk
Please enter product category:Dairy
Please enter number of products to add:5
Please enter the price of the product:4.5
Please enter expiration date of the product[dd/mm/yy]:12/05/20
The product Milk -barcode:0123456789 ,added successfully
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:9876543210
Please enter product name:Bread
Please enter product category:Pastries
Please enter number of products to add:6
Please enter the price of the product:2.50
Please enter expiration date of the product[dd/mm/yy]:20/05/20
The product Bread -barcode:9876543210 ,added successfully
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:0521369874
Please enter product name:Sweet Chilli
Please enter product category:Sauces
Please enter number of products to add:6
Please enter the price of the product:20
Please enter expiration date of the product[dd/mm/yy]:20/06/20
The product Sweet Chilli -barcode:0521369874 ,added successfully
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:3
What date you want to check[dd/mm/yy]:25/05/20
~~~~~~~~~~~~~~~Expired Products~~~~~~~~~~~~~~~
Product name: Milk
Product barcode: 0123456789
Product expiration date: 12/5/20
Product name: Bread
Product barcode: 987654321
Product expiration date: 20/5/20
```

4. Print all the products

If the user chooses this option, then all the products currently in the store should be printed. If there are products, you should print "No products in the store!".

```
Example:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:04568324
Please enter product name:Apple
Please enter product category:Fruit
Please enter number of products to add:3
Please enter the price of the product:8
Please enter expiration date of the product[dd/mm/yy]:06/06/20
The product Apple -barcode:04568324 ,added successfully
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:1
Please enter product barcode:1234
Please enter product name:Carrot
Please enter product category:Healty
Please enter number of products to add:80
Please enter the price of the product:4
Please enter expiration date of the product[dd/mm/yy]:20/05/20
The product Carrot -barcode:1234 ,added successfully
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
```

```
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:4
~~~~~~~~~~~~~~~All Products~~~~~~~~~~~~~~~
----------
Product name: Apple
Product barcode: 04568324
Product category: Fruit
Product available quantity: 3
Product price: 8
Product expiration date: 6/6/20
----------
Product name: Carrot
Product barcode: 1234
Product category: Healty
Product available quantity: 80
Product price: 4
Product expiration date: 20/5/20
Total number of products: 2
```

5. Update product

The user can ask to update the product details. It should be possible to update all the details except for the product barcode.

If the store is empty you shouldn't ask for the barcode, just print:

printf("No products in the store!\n");

If the barcode can't be found (because there is no product with this barcode) type, and ask for the barcode again until right barcode was typed:

printf("Couldn't find the product barcode, try again...\n");

Example:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:5
Please enter product barcode you want to update:1111
Couldn't find the product barcode, try again...
Please enter product barcode you want to update:1234
What do you want to update?
        1. Product name
        2. Product category
        3. Product quantity
        4. Product price
        5. Product expiration date
Please choose operation [1-3]:3
Please enter new product quantity:50
```

6. EXIT SYSTEM

In this case, the program should free all the memory that was allocated and stop the execution.

Example:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Welcome to CORONA market!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Manage market menu:
        1. Add product
        2. Remove product
        3. Check which products are expired
        4. Print all the products
        5. Update product
        6. EXIT SYSTEM
Please choose operation [1-6]:6
exit...
```

Skeleton prints:

Use this print code to make your code more correct and easier to read.

Interface print:

```
printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");
printf("Welcome to CORONA market!\n");
printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");
printf("Manage market menu:\n");
printf("    1. Add product\n");
printf("    2. Remove product\n");
printf("    3. Check which products are expired\n");
printf("    4. Print all the products\n");
printf("    5. Update product\n");
printf("    6. EXIT SYSTEM\n");
printf("Please choose operation [1-6]:");
```

(Don't worry about the tab space, it should look good when you copy-paste it)

Update product interface:

```
printf("What do you want to update?\n");
printf("    1. Product name\n");
printf("    2. Product category\n");
printf("    3. Product quantity\n");
printf("    4. Product price\n");
printf("    5. Product expiration date\n");
printf("Please choose operation [1-5]:");
```

Important Notes:

- In order to read a string with spaces until the new line (useful for the product name and category name), use this scanf code:
  `scanf("\n%[^\n]s",string);`
- Always check if memory allocation succeeds even if the chance of it to fail is small. It will be easier to debug using prints.
  Example:
  ```
  Product * ptr = NULL;
  ptr = malloc(sizeof(product));
  if (ptr==NULL)//memory allocation failed
  {//do something}
  ```
- Every function needs to be documented in this fashion:
  ```
  /*Inputs:(arguments that the function receives)
  Return parameter:(if the function is void write None)
  ```

```
Function functionality: (1-3) sentences what is the purpose of
this function and how you're going to implement this
functionality*/
```

- Remember, it's always better to have 3 functions of 50 lines then one of 150 lines. Try to split functions with different functionality. If possible, keep the main function as clean as possible.
- It's forbidden to add fields to the given structs or implement new structs.
- You can always assume that the input is valid and with a valid length.
- <u>Your output should be 100% the same as the output example given above.</u>

Submission rules

This is the first big project you are doing this semester. We are aware of its difficulty but it is meant to prepare you well for the final exam.

1) **This homework is supposed to be done in pairs.**
2) Deadlines:

   Upload time: 18:00 on 7.5.

   Submission deadline: 23:59 on 28.5.

   It is okay to work alone, but the difficulty will remain the same.

   Only one of the pair has to submit the assignment.
3) C file submission pattern:

   if submitted in pairs: **ex4_id1_id2.c**

   if submitted alone: **ex4_id1.c**
4) The .c file mentioned above should be submitted in .zip file in this manner:

   if submitted in pairs: **ex4_id1_id2.zip**

   if submitted alone: **ex4_id1.zip**

   id1 and id2 are the respective student id's

   Good luck to everybody :)