Program Persist Data: Group Assignment1

Even Teams (team numbers 2, 4, 6, 8 &10 from all lab groups)
General Instructions
Prepare and submit before April 22nd at 5pm.
Equal marks for each task.
This will be demoed in lab and to the class within the teams.

Prepare using C, error check and comment all 'moving parts' appropriately[1]. You are given directions create function names that are appropriate and use functions wherever possible. Reference any code that you use that you have not created or been given in class.
Consider getting a GitHub account and work within a project space for each task.

1. Scramble a File

You are required to create two utilities; scramble.c and unscramble.c that are called by command line specifying the name of the file to scramble/unscramble.
For example:

```
C:> scramble myfile.txt
C:> unscramble myfile.txt.sbl
```

The scramble utility reads a file (any type of file) and generates a new file with the same name as the input file but with the extension .sbl added to the end of the file (keep the previous extension, for instance picture.bmp becomes picture.bmp.sbl). The .sbl file is generated by swapping bytes two by two. For instance, the 1stbyte is swapped with the 2nd, the 3rd with the 4th and so on..

For instance if the input file is

byte 1	byte 2	byte 3	byte 4	...	byte n-1	byte n

The output file will be:

byte 2	byte 1	byte 4	byte 3	...	byte n	byte n-1

If the input file has an odd number of bytes the last one will not be swapped. The scramble utility encrypts a file so that it cannot be read anymore by the right application. The utility unscramble does the reverse task: it takes a scrambled file.sbl and recreate the original file by removing the .sbl extension from the name and reswapping back the bytes. Try the program with any type of file and check if, by scrambling and unscrambling a file, the resulting file is the same as the original starting file.

2. Phonebook implementation

Create a simple phonebook that performs the following functions:

1. Add new contact
2. Delete old contact
3. Edit contact
4. Find contact
5. Record contacts

Sample contact:

Name	Phone	email
Jane	4558	jane.ferris@dit.ie

[1] Over commenting is essential in team exercises.

This program is to be implemented using an array. The code should use functions where appropriate and effective error checking. Select an appropriate unique identifier (key) assume that no contacts share a phone.
For the fifth function 'Record contacts' implement an appropriate sorting algorithm based on the design of a phone book of max 200[2] contacts and create a file called phonebook that is used to store the phonebook for reference. An additional variable within the file or filename argument will be required to date the phonebook records file.

In addition to the required user functions, create an additional display() in order to demonstrate your code after each addition, deletion or alteration. When the phonebook is full send an alert to the user that no further contacts may be added.
In order to demonstrate use an array of 5 with 3 pre-initialised contacts.

3. **Queue implementation.**

Using a linked list dynamic data structure implement a queue structure IT services ticketing system.
The system will enable the user add tickets when IT fix requests are received via phone.
The important information for the system is as follows:

Name of client:
Contact Number:
Short description of issue:
Time of phone call:

The ticketing system will only allow the next ticket in the queue be processed and the queue has a limit of 5 tickets. Select an appropriate unique identifier for the system[3].

Please initialise 3 tickets in whatever manner you desire in order to demonstrate your system.

Jane	4558	Monitor broken	12.05
Sean	5656	Desktop won't flickering	12.06
Mark	1212	Router lights amber	12.10

Code the system using functions where possible and create an additional display() in order to demonstrate your code after each new input to the queue and each dequeueing (processing of tickets at start of queue). When the queue is full send an alert to the user that no further tickets may be taken.

[2] Use a symbolic constant so you may alter to demo.
[3] The IT office has 1 phone between departments, their clients have a hot desk environment which means office phone is a shared resource.