Foundation Year in Computing Sciences

Foundation Programming – G6065

Autumn term 2020 – Programming Project 2 – Covid-19 MXA Trial

Set:	Thursday 19 November 2020
Due:	Now due 17 December – E-Submission
Format:	A single word processed document.
Learning Outcomes Assessed	1. Employ a range of basic programming constructs to develop a programming solution in a suitable high-level, imperative programming language 2. Transform a program specification into a design using a standard top-down design technique 3. Use a programming environment to edit, debug and compile a simple program 4. Design and use a test plan for verification of a program, and draw conclusions from the outcomes

Specification and Tasks:

You are employed as a junior programmer by Cheylesmore Software Focus, a small software house that produces bespoke software for a range of clients. Sabrina, your line manager, has asked you to write a small system of two C programs for a local research medical team "Covid-19 MXA Trial".

The charity has asked for two programs to be written. First, design, write and test a program that allows one of their volunteers to enter trial participant details into a computer file prior to the trial participant being seen by a research doctor for the vaccine injection. Second, design, write and test a program that allows the research doctor to read and update a trial participant's files. The follow up system is to be written by another team.

The five volunteers are Bill Gates, Boris Johnson, Mark Drakeford, Nicola Sturgeon, and Arlene Foster. They have been assigned the task of entering the trial participants' details. Only these five volunteers will be allowed to access the program that you are about to write. No one other than these five volunteers should be allowed to use the system, so each volunteer needs to be given a unique login id and password. Ids and passwords start with a letter and contain only letters and numerical digits. Passwords are to be exactly 7 characters in length. To add to the security the program is to store only encrypted versions of each password in an array in memory. To encrypt a password add 7 to each character code in the password, wrapping around from 'z' to 'a' and '9' to '0', eg password "aBc2xyz8" when encrypted becomes "hLj9efg5". The ids and encrypted passwords should be stored in arrays in memory.

Data written to a trial participant's file should be:

- First name
- Last name
- Date of birth
- Height
- Weight
- Medical conditions

- Current medication
- Vaccine code number (a unique 9 digit number)
- Additional comment

Each trial participant's details will be written to a file and encrypted using the same algorithm that was used for the passwords. Punctuation and spaces remain unencrypted. The name of the file will be the trial participant's last name, plus their date of birth as 6 digits with an extension of ".mxa" eg "smith010285.mxa".

The research doctor, Michel Barnier or David Frost, should have a password to log into the second program. After logging in Michel or David should enter the trial participant's last name and date of birth. The program will then read the appropriate file and present the trial participant's details in a readable form on the screen. The research doctor should then enter the vaccine code number and any additional comments to the trial participant's notes before resaving the data to the file.

Additionally, when either program starts, the stored ids and encrypted passwords should be read from a password file. Any maintenance of ids or passwords should be done by the research doctor in the second program. The research doctor should be allowed to add, delete or change any volunteers' id or password. When the research doctor's program finishes the password file should be overwritten with the updated stored ids and encrypted passwords.

What to submit:

A single word processed document containing all tasks attempted listed below.

The tasks required are:

1. A word processed document containing the design, the C programming code, the test logs and a written evaluation of the program. A contents page using appropriate page numbering should be included. (8 marks)
2. For both programs the design includes screen layouts, data storage design of arrays and files, a test plan, and skeleton code at level 1 and 2. (10 marks)
3. Program 1 and program 2 have basic functionality, not necessarily with any encryption or a separate password file. (10 marks)
4. Data encryption and password encryption work appropriately. (10 Marks)
5. Encrypted passwords are stored in a maintained password file. (14 Marks)
6. Small appropriate user-defined functions are used, with the layout conforming to the course's "Style Rules for C Programming". (14 marks)
7. Four test logs are provided that use the data specified in the design test plan with additional tests added as appropriate. The logs must show how the programs deal with invalid data, and what works and what does not work. (16 marks)
8. A good user interface with clear prompts, where items are set out neatly in rows and columns. Additionally box characters may be used. (8 marks)
9. The written evaluation with a word count (500 words). This should clearly describe what works and what does not work in the program. It should also outline the skills you used, highlighting what research or help was needed for the various sections of the project. (10 marks)