Birzeit University

Department of Electrical & Computer Engineering

Summer Semester, 2023/2024

ENCS3130 Linux Laboratory

Shell Scripting Project – Medical Test Management System

Objective:

Develop a system in shell scripting to efficiently store, manage, and retrieve medical test data for individual patients. This system acts as a basic patient record management system focusing on test results. The system will include features for adding new test results, updating existing records, deleting outdated or incorrect entries, and retrieving specific test results based on patient ID or test type.

File Format:

The medical test will be stored in text file named midecalRecord. Each line in the text file represents a single medical test. The representation will include fields for:

- Patient ID (integer: 7 digits)
- Test name (string - consider a fixed length)
- Test date (string - fixed format like YYYY-MM)
- Result (floating-point value with unit (string - fixed length))
- Status (string - fixed length, e.g., "Pending", "Completed", "Reviewed")

For example, the following file has two medical tests:

```
1300500: RBC, 2024-03, 13.5, mg/dL, completed
1300511: LDL, 2024-03, 110, mg/dL, pending
```

Medical Tests:

There is another file that store the information of each test. Below is the list of medical tests with their normal range:

1. Name: Hemoglobin (Hgb); Range: > 13.8, < 17.2; Unit: g/dL
2. Name: Blood Glucose Test (BGT); Range: > 70, < 99; Unit: mg/dL
3. Name: LDL Cholesterol Low-Density Lipoprotein (LDL); Range: < 100; Unit: mg/dL
4. Name: Systolic Blood Pressure (systole); Range: < 120; Unit: mm Hg
5. Name: Diastolic Blood Pressure (diastole); Range: < 80; Unit: mm Hg

The format of the second file named medicalTest contains four information about each test: name, range, unit. This file will be stored in the management system as a text file containing all medical tests offered by the clinic/hospital.

System Functionality:

Develop a text-based menu that allows users to:

- Add a new medical test record: the system will allow the user to store a new medical test with the required data. The system will check the validity of the input data.
- Search for a test by patient ID: the system will have the following functionality based on user selection:
    - Retrieve all patient tests
    - Retrieve all up normal patient tests
    - Retrieve all patient tests in a given specific period
    - Retrieve all patient tests based on test status
- Searching for up normal tests: the system will retrieve all up normal patients' tests based on the input medical test.
- Average test value: the system will retrieve the average value of each medical test.
- Update an existing test result.

In addition to the above functionality, the system has capability for:

- **Error Handling:** Implement error handling for invalid file name, searching for non-existent tests, searching for non-existent patient, ….
- **Data Validation:** Validate user input to ensure proper data types (e.g., integers for ID, valid dates) and handle potential errors.

Submission:

Please submit the following:

1. Code
2. At least 2 testing examples.

Notes:

- Write the code for the shell program to satisfy the requirements described above and name the file as MedicalTest.
- Make sure your code is clean and well indented; variables have meaningful names, etc.
- Make sure your code has enough comments inserted to add clarity.
- Work in groups of at most two students
- Deadline: **Monday,** 12 **August, 2024 at 11:59pm.** Please submit your project (code + test cases) through Ritaj as a reply to this message.
- This project is per group effort: instances of cheating will result in you failing the course.

Grading Policy:

Item	Points
Add new medical test record and Data Validation	30
Search for a test by patient ID	30
Searching for up normal tests	20
Update an existing test result	10
Delete a test	10
Average Value	10
Error Handling	10
Code structure	10
Discussion	20
Total	150