Pet Assignment: COBOL (Joe and Mubarak)

Submit To: jpfielding@loyola.edu, moayantayo@loyola.edu

Goal

This assignment explores COBOL's (COmmon Business Oriented Language) excellent file processing capabilities. You have done a lot of programming with files treated as a sequence of bytes. Now you will be working with indexed files, which allow you to conveniently access file records by a key (think hashtable). This assignment will also involve fixed point numbers rather than floating point numbers. You will be able to appreciate how we can use fixed point numbers in COBOL to avoid potential calculation inaccuracies that come with using floating point numbers.

Introduction

On the potter box, here is how you can compile a COBOL program:

```
/usr/local/bin/cobc file.cbl -x -o out
```

This creates an executable named out

Here are instructions for installing COBOL locally on linux:

```
wget http://sourceforge.net/projects/gnucobol/files/gnucobol/3.2/gnucobol-3.2.tar.gz
```

```
tar xvf gnucobol-3.2.tar.gz
```

```
cd gnucobol-3.2
```

```
./configure
```

```
make
```

More information: https://sourceforge.net/projects/gnucobol/files/

This assignment contains 2 COBOL files: practice.cbl and handout.cbl. practice.cbl is just for you to practice with, and handout.cbl is the base for your solution.

Additionally, we have provided 2 data files: practice.dat and students.dat. You can test the practice program with practice.dat, and test the program you will submit with students.dat

Requirements

Your task is to complete the COBOL program in handout.cbl (which is partially implemented). This program will ask the user for the name of a file. This file will be an indexed file where each record contains 40 characters: 8 characters for the student's name (this is also the key used to look up the record), followed by 32 digits representing 8 grades (4 digits per grade). The first 2 digits of the grade come before and the last 2 digits come after the decimal place (ABCD, where A, B, C, D are digits represents AB.CD). The program will repeatedly ask the user for a name and print out the average of that student's 8 grades, or display a message saying that the student is not in the file. The program will stop asking for names when the user enters 0. After this, the program should print the frequency at which each letter grade (A,B,C,D,F) was achieved. These frequencies only include students that the user asked for during the previous part of this program Do not worry about handling the case where the user enters the same student twice. Additionally, do not worry about boring input validation. Each place in handout.cbl where you need to add code will be marked with a TODO comment.

Steps

Step 0: Practice (Optional)

The file practice.cbl is meant just as a guide to help you get used to programming in COBOL. This program asks the user for a file containing student grades, where each line is 1 record containing the student's name as the first 8 characters, and their grade as the last 4 digits (same format as described in the requirements section). It then repeatedly asks the user for entries to append to the file until the user decides to quit. It will then print out the average grade of all the students. This program already works as intended, but feel free to play around with it for practice. You will not need to submit this file.

Step 1: Complete handout.cbl

Complete handout.cbl so that it works as described in the requirements section, follow the steps below for our recommended way of completing this assignment.

Step 2: Define File Records

In the FILE SECTION, define the structure of the records in the indexed file (as described in the requirements section). We recommend using the OCCURS clause rather than making 8 separate fields for the grades. Note that you will have to define this same structure in the WORKING-STORAGE SECTION so that you can load in each record for processing.

Step 3: GET-ENTRIES paragraph

Complete this paragraph so that it repeatedly asks the user for a student's name until they quit. You will have to add variables to the working storage section as you write this paragraph.

Step 4: LOOKUP-STUDENT paragraph

Write this paragraph so that it searches for the student's record using the name as they key. It should do all the necessary processing as described in the requirements if the student is in the file, but if they are not, display a message saying so.

Step 5: PROCESS-RECORD paragraph

In this paragraph, you should calculate the student's average grade and update the appropriate letter grade frequency.

Step 6: AVERAGE paragraph

Calculate the average grade out of the student's 8 grades.

Step 7: DISPLAY-FREQUENCIES paragraph

This paragraph should display each of the letter frequencies in a human-readable way.

What To Hand In

Your completed program in handout.cbl

Data File:

practice.dat

practice.dat includes 1 record per line, where the first 8 characters are the student's name, and the last 4 digit are their grade

```
 Alice   9467
Bob     1244
Carlie  7358
```

students.dat:

students.dat contains these students and their corresponding 8 grades

```
 Alice  : [90.00, 81.00, 90.00, 96.00, 90.00, 90.00, 90.00, 89.00]
Bob    : [60.00, 90.00, 70.00, 60.55, 98.89, 56.67, 90.00, 80.00]
Charlie: [91.00, 70.00, 59.00, 60.00, 75.55, 64.32, 79.87, 73.32]
Dan    : [60.00, 60.00, 60.00, 60.00, 60.00, 60.00, 60.00, 59.94]
Eve    : [99.99, 99.99, 99.99, 99.99, 99.99, 99.99, 99.99, 99.99]
```

If the user requests the average grade of each student, this should be the output (assuming they do not enter any student more than once):

```
 A: 1
B: 1
C: 2
D: 0
F: 1
```