Curtin College

End of Study Period Practice Assessment

Unit: UCP1000 Unix and C Programming

THIS IS AN OPEN BOOK ASSESSMENT

You are required to write C code that meets requirements outlined in the following tasks.

You are allowed to compile and test your code as you desire. However, it is your responsibility to ensure that you have a working environment you can test in.

How to Submit

You do not need to submit this practice assessment. However, in the final, your submission is to consist of a single `.tar.gz` file that contains all the .c, .h, .sh and makefile files you have written for the assessment.

Input Files

The following example input files for testing are available to download from Moodle.

- gold_street_data.txt
- hendworth_mall_data.txt
- mega_island_data.txt

Overall Program

SuperFun Entertainment is a movie theatre company that owns theatres around Australia. They have recently been bought by a large conglomerate and have commissioned you to write a C application to process movie data. Each cinema contains a collection of movies with times and dates for when they will be shown. Your program is to read a collection of these movies from text files into one or more **Linked List** data structures and perform the required processing operations on the data. Your program should be compiled with make and contain a working makefile.

For this program, refer to the following standard C function prototypes:

```
FILE *fopen(const char *path, const char *mode);
int fclose(FILE *fp);
int fscanf(FILE *stream, const char *format, ...);
char *fgets(char *s, int size, FILE *stream);
int fgetc(FILE *stream);
int feof(FILE *stream);
int sscanf(const char *str, const char *format, ...);int
strcmp(const char *s1, const char *s2);
char *strncpy(char *dest, const char *src, size_t n);
int atoi(const char *nptr);
```

Task 1 – Data Types (15 Marks)

Declare suitable C datatypes to represent each of the following sets of information in a header file.

1. A **Movie** type that includes
    - Movie ID (Integer Value)
    - Movie name (max 127 characters)
    - Date (three integers day/month/year)
    - Time (two integers hour:minute)

2. A **Cinema** can include a collection of one or more **movies** and a cinema id (an integer)
   Hint: Marks have been allocated for using a **linked list**

3. A **Theatre** that has a name (max 127 characters), one or more **cinemas** and a count (an integer)

Task 2 – Loading Data (25 Marks)

Write a function (or collection of functions) that will read a list of movies from text files and populate a SuperFun Theatre.

SuperFun Theatre data is saved in text files.

The name of the theatre is shown on the first line of the file. After that, each line of the file will include information about the cinemas and movies. The Cinema is listed with its cinemaID followed by all the movies shown in that cinema.

The Movie data in the file is delimited by spaces in the following format:

```
day/month/year hour:minute movie name
```

The following is an example theatre file (Available to download on Moodle):

"gold_street_data.txt"

```
Gold Street Theatre
Cinema 1
1/1/2021 12:20 Sarnac the HorgeHeg 2
1/1/2021 15:45 Raiders of the Last Orc
1/1/2021 19:30 Tin Man Legacy
2/1/2021 12:20 Sarnac the HorgeHeg 2
2/1/2021 15:45 Raiders of the Last Orc
2/1/2021 19:30 Tin Man Legacy
3/1/2021 12:20 Sarnac the HorgeHeg 2
3/1/2021 15:45 Raiders of the Last Orc
3/1/2021 19:30 Tin Man Legacy
Cinema 2
1/1/2021 12:20 Stare Wares Episode XII
1/1/2021 15:45 Detective Rabbit 3D
1/1/2021 19:30 Tin Man Legacy
2/1/2021 12:20 Sarnac the HorgeHeg 2
2/1/2021 15:45 Detective Rabbit 3D
2/1/2021 19:30 Tin Man Legacy
```

Your function should:

- Take in a filename
- Dynamically allocate the structures you designed in Task 1.
- Read each **movie** and store it into an appropriate **cinema**.
- Return a pointer to a single theatre structure you designed in Task 1.

If the file cannot be opened, your function should return NULL. An error message is not required.

Task 3 – Random Cinema (15 Marks)

Write a function (or collection of functions) that will import a pointer to a theatre. This function should select a random cinema from the theatre using a seeded random number generator.

For every movie within the randomly selected cinema, the function should output the name of the movie.

The ID of the cinema should then be doubled four times (using binary manipulation operations).

For example, calling:

```
randomCinema(theatre);
```

Would select a random Cinema within the provided theatre and output the movies, before updating the cinema's ID. If "Cinema 2" was selected, it would become "Cinema 32"

Task 4 – Main (15 Marks)

Write a main function that accepts any number of filenames (at least 1) as command-line parameters.

Your main function should:

- For each filename provided:
  - Use the loading code from Task 2 to read the contents of the file into a theatre.
  - Use the code from Task 3 on the data.
- If a theatre fails to read, print an error message and return 1, otherwise return 0
- Perform all necessary freeing of dynamically allocated memory.

Task 5 – Shell Script (10 Marks)

Write a bash script named movies.sh that will perform the following operations in order.

1. Use the grep command to search through all the .c source code and display any lines that end with a `';'` character
2. Use make to compile your program
3. Execute your program using the three input files provided on Moodle
4. Execute your program a second time but redirect the program's output to a file named "output.txt"

If your program does not return 0 when executed in steps 3 or 4, the bash script should echo an error message.

Compilation and Makefile (10 Marks)

Your code should compile using make and should have a well-structured and functioning makefile.

Coding Practices (10 Marks)

Your code should be well structured and easy to read. As with any code for this unit, it is to abide by the Coding Standards that have been set from the start.