Hacettepe University
Department of Computer Engineering
BBM203 Programming Laboratory
Experiment 1

Subject : A basic cinema automation with C
Submission Date : 17.10.2016
Due Date : 30.10.2016
Programming Environment: ANSI C / MinGW / Windows
Advisors : Dr. Sevil ŞEN, Dr. Burcu CAN, Dr. Adnan ÖZSOY
Dr. Ahmet Selman BOZKIR

AIM

In this experiment you are expected to design and implement a basic console application that simulates a real cinema automation. The main goal of this assignment is to encourage students to build matrix like data structures by using dynamic memory allocation. Meanwhile, you will have a chance for reviewing and learning concepts such as pointers and structures in C.

WARMUP

Theaters in all over the world are generally being managed by utilizing specialized three tiered client/server type applications. Moreover, those applications have been powered by enabling *web services* for external systems. In this assignment, you are supposed to create a very basic version of this type of applications. You will develop your application in console application format. Basically, your application will read a text document (name of the file will be passed as an argument) and save the output of the processed commands. Output should be generated for both screen and text file. During your coding activities, you are **prohibited** to use C++ libraries and syntax. ANSI C must be employed during implementation.

DETAILS - COMMANDS

In this experiment you are supposed to design and implement a console based ticket reservation application. All commands will be stored in a text file which will be processed by your application.

Your application will involve 4 commands: CREATEHALL, BUYTICKET, CANCELTICKET, SHOWHALL and STATISTICS. A two dimensional dynamic sized hall will be created and every seat in a hall will have a label (i.e. G10). During purchasing a seat can be sold as either a student or as a full fare. Price of student fare is 7 TL and full fare is 10 TL (these prices are static and will not change). Further details belonging to commands were listed below. Note that, **command names are case sensitive** and commands will be error free in terms of syntax.

CREATEHALL:
This command creates a hall with supplied parameters. Command prototype is given below:

```
CREATEHALL "Hall_name" "Movie_name" HallWidth HallHeight
             String       string       int       int

Sample Command:
CREATEHALL "Red_hall" "Interpreter" 28 20
Output:
N/A
```

Note that, in our sample application, hall width cannot exceed 28 since the English alphabet contains 28 characters. Therefore the first seat (referring to x axis) in the hall is denoted as 'A', second one is 'B' and so on so forth. However, regarding the columns, seats are labeled as 1, 2, 3… For the sake of simplicity, the top leftmost corner is labeled as A1.

Hall or movie names cannot contain whitespace and they are written between double quotes. Halls are created dynamically therefore you **must** benefit from dynamic memory allocation. On the other hand, each seat is initially free and ready to be purchased.

BUYTICKET:
This command is used for selling tickets. Ticket price may change depending on two categories: (1) Student and (2) FullPrice. Ticket prices are 7 and 10 Turkish Liras for students and adults respectively. Seat labels represent the column and row positions of a seat. A buyer can purchase multiple seats (adjacent to each other) at a time. Thus, you must first check whether requested seat(s) are defined in that hall. Moreover, one or multiple seats may not be available due to the prior purchase of someone. Therefore, you must also check whether the requested seat(s) are available to be sold. For instance, a person may request E6, F6, G6. Even if one of the requested seats is already sold, then you cannot sell it partially.

Command prototype is given below:

```
BUYTICKET "Movie_name" SeatLabel TicketType #ofSeatsRequested
             String      String     String          int

Sample Command:
BUYTICKET "Last_Samurai" D4 Student 2

Successful Output:
Orange-Hall [Last_Samurai] Seat(s) D4,E4 have been sold.
Possible Error Messages
ERROR: Seat D4 is not defined at "Orange-Hall"
ERROR: Specified seat(s) are not available! They have been already taken.
```

CANCELTICKET:
For cancellation purposes, the command "CANCELTICKET" has been designed. For ticket cancellations your application must handle this command that its syntax has been given below. Note that, if the seat about to be cancelled was not purchased before, raise an error message.

```
CANCELTICKET "Movie_name" SeatLabel
                String      String

Sample Command:
CANCELTICKET "Last_Samurai" D4
```

Successful Output:

```
Orange-Hall [Last_Samurai] purchase is cancelled. Seat D4 is now free.
```

Possible Error Messages

```
ERROR: Seat D4 in "Orange-Hall" was not sold.
```

SHOWHALL:

This command is used in order to display the current status of the seats to visualize the seating plan of the hall.

```
SHOWHALL "Hall_name"
          String
```

Sample Command:

```
SHOWHALL "Orange-Hall"
```

Successful Output:

```
Orange-Hall sitting plan
  ---------------------
1 | | | | | | | | | | |
  ---------------------
2 | | | | | | | | | | |
  ---------------------
3 | | | | | | | | | | |
  ---------------------
4 | | |f|f| | | | | | |
  ---------------------
5 | | | | | | | | | | |
  ---------------------
6 | | | | | | | | | | |
  ---------------------
7 | | | | | | | | | | |
  ---------------------
8 | | | | | | | | | | |
  ---------------------
9 | | | | | | | | | | |
  ---------------------
10| | | | | | | | | | |
  ---------------------
11| | | | | | | | | | |
  ---------------------
12| | | | | | | | | | |
  ---------------------
13| | | | | | | | | | |
  ---------------------
14| | | | | | | | | | |
  ---------------------
15| | | | |s|s| | | | |
  ---------------------
16| | | | | | | | | | |
  ---------------------
17| | | | | | | | | | |
  ---------------------
18| | | | | | | | | | |
  ---------------------
19| | | | | | | | | | |
  ---------------------
20| | | | | | | | | | |
  ---------------------
   A B C D E F G H I J
     C U R T A I N
```

As you can see from the output depicted above, SHOWHALL command prints out the seating plan including the screen. Note that, the seating plan must be visualized exactly in the same format with this command.

STATISTICS:
In order to learn and display which movies make money, STASTICS command will be used. This command does not take any parameter and it only outputs the created movies along with the total ticket cost.

Sample Command:
STATISCTICS
Successful Output:

```
Last_Samurai 5 student, 2 full fare, sum: 55 TL
Inception 8 student, 4 full fare, sum: 76 TL
London_has_fallen 6 student, 3 full fare, sum: 57 TL
```

NOTES

- Input file will be in the same directory with your executable files. Therefore prepare your code to search the input file by its name (via argument) and process it.
- SAVE all your work until the experiment is graded.
- The assignment **must be original**, INDIVIDUAL work. Downloaded or modified source codes will be considered as cheating. Also students who share their work will be punished in the same way.
- You can ask your questions via course's piazza group. However, I suggest that you write your questions in English instead of Turkish.

SUBMISSONS

- The experiment code will be tested both MinGW and dev.cs.hacettepe.edu.tr. Please ensure that your code is well working and compliant with ANSI C standart.
- Your submission will be in the format below You must submit your code file in this format. For the process of obtaining the zip file, you can use Windows folder compression technique.
  <BBM203_1617_1_StudentID>
  |-- source
  |-- All your solution folder

- Cheating is definitely prohibited. Students who are responsible for this kind of plagiarism will be punished by the rules of Hacettepe University. Therefore, if you want to cheat someone else code, think twice.
- You have to use "Online Experiment Submission System". **http://submit.cs.hacettepe.edu.tr** Other type of submissions especially by e-mail WILL NOT BE ACCEPTED.
- Submission deadline is 30.10.2016 23.59 pm. **No further extension will be given!!!**