CS162 – Programming Techniques

Lab 04
Pointer (cont.)

Cảm ơn thầy Trần Duy Quang đã cung cấp template cho môn học

Department of Software Engineering-FIT-VNU-HCMUS

1

Notes

Create a single solution/folder to store your source code in a week.

Then, create a project/sub-folder to store your source code of each assignment.

The source code in an assignment should have at least 3 files:

- A header file (.h): struct definition, function prototypes/definition.
- A source file (.cpp): function implementation.
- Another source file (.cpp): named YourID_Ex01.cpp, main function. Replace 01 by id of an assignment.

Make sure your source code was built correctly. Use many test cases to check your code before submitting to Moodle.

Name of your submission, for example: **18125001_W04_03.zip**

2

Content

In this lab, we will review the following topics:

- How to use a pointer with Structure.

3

Assignments

A: 1 problems / assignments.

H: all problems / assignments.

3.1 Assignment 1

Using **pointer and dynamic array** to write a program that can do the following functions:

- Read from keyboard an array of N fractions.
- Extract negative fractions to another array.
- Display the list of negative fractions you found on screen.
- Find the sum of positive fractions and display that result on screen.

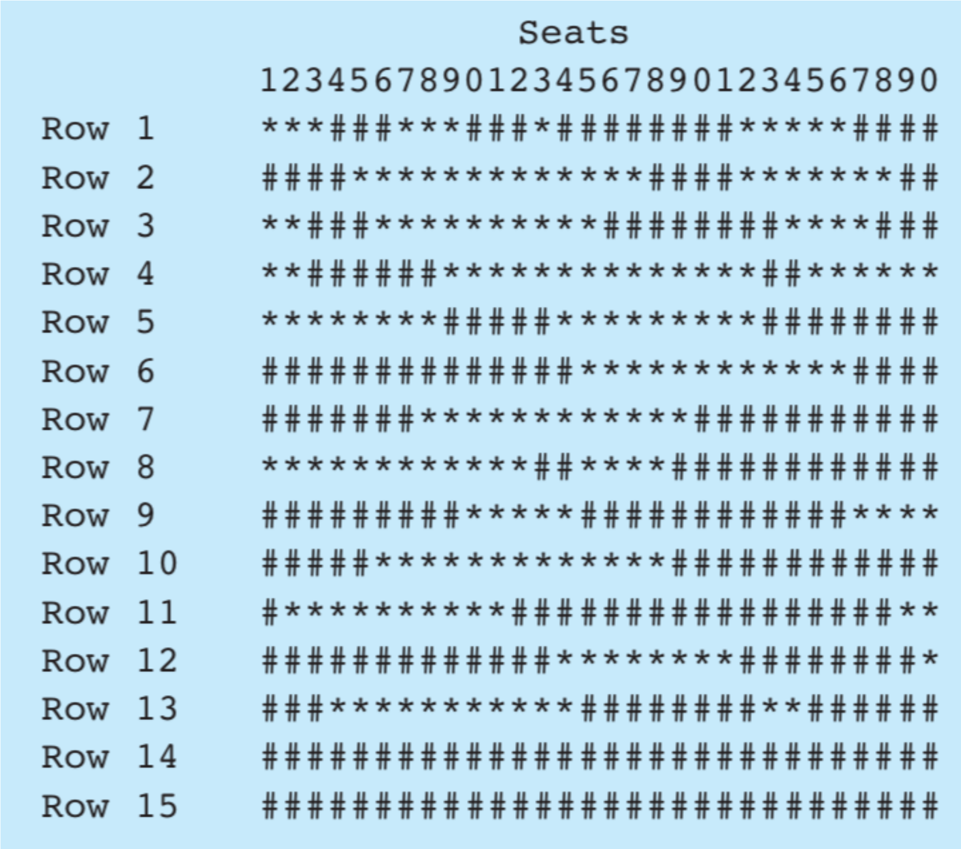
3.2 Assignment 2

Using **pointer and dynamic array** to write a program that can do the following functions:

- Read from keyboard an array of N students
  - ID (int)
  - Name (char*)
  - Score (Double)
  - Address (char*)
- Find the list of students whose scores are less than 5 to an array.
  - Display the list you found above on screen.
- Sort and display the list of students in ascending order based on their ID
- Sort and display the list of students in descending order based on their name
- Find top 3 students who have the highest score
  - Display the list you found above on screen.

3.3 Assignment 10

Write a program that can be used by a small theater to sell tickets for performances. The theater's auditorium has 15 rows of seats, with 30 seats in each row. The program should display a screen that shows which seats are available and which are taken. For example, the following screen shows a chart depicting each seat in the theater. Seats that are taken are represented by an * symbol, and seats that are available are represented by a # symbol:



```
                        Seats
           123456789012345678901234567890
Row  1     ***###***###*########*****####
Row  2     ####*************####*******##
Row  3     **###**********########****###
Row  4     **######**************##******
Row  5     ********#####*********########
Row  6     ##############************####
Row  7     #######************###########
Row  8     ************##****############
Row  9     #########*****############****
Row 10     #####*************############
Row 11     #**********#################**
Row 12     #############********########*
Row 13     ###***********########**######
Row 14     ##############################
Row 15     ##############################
```

Here is a list of tasks this program must perform:

- When the program begins, it should ask the user to enter the seat prices for each row. The prices can be stored in a separate array. (Alternatively, the prices may be read from a file.)
- Once the prices are entered, the program should display a seating chart similar to the one shown above. The user may enter the row and seat numbers for tickets being sold. Every time a ticket or group of tickets is purchased, the program should display the total ticket prices and update the seating chart.
- The program should keep a total of all ticket sales. The user should be given an option of viewing this amount.
- The program should also give the user an option to see a list of how many seats have been sold, how many seats are available in each row, and how many seats are available in the entire auditorium.

Input Validation: When tickets are being sold, do not accept row or seat numbers that do not exist. When someone requests a particular seat, the program should make sure that seat is available before it is sold.