Lab 03
Dynamic Arrays

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_W01_07.zip**

2

Content

In this lab, we will review the following topics:

- How to use a pointer as a 1D array.
- How to use a pointer as a 2D array.

3

Assignments

A: YY: 02 problems / assignments.

H: YY: 06 problems / assignments.

- **Assignment 1, 2: choose 1.**
- **Assignment 3.**
- **Assignment 4, 5, 6, 7: choose 2.**
- **Assignment 8, 9, 10: choose 1.**
- **Assignment 11.**

3.1 Assignment 1

In the C programming language there is no pass-by-reference syntax to pass a variable by reference to a function. Instead a variable is passed by pointer (just to be confusing, sometimes passing by pointer is referred to as pass by reference). This Practice Program asks you to do the same thing as C, which in practice would be simpler to implement using C++'s reference parameter syntax. Here is the header for a function that takes as input a pointer to an integer:

```
void addOne(int *ptrNum);
```

Complete the function so it adds one to the integer referenced by ptrNum. Write a main function where an integer variable is defined, give it an initial value, call addOne, and output the variable. It should be incremented by 1.

3.2 Assignment 2

The following function uses reference variables as parameters. Rewrite the function so it uses pointers instead of reference variables, and then demonstrate the function in a complete program.

```
int doSomething(int &x, int &y) {
        int temp = x;
        x = y * 10;
        y = temp * 10;
        return x + y;
}
```

3.3 Assignment 3

This Practice Program requires that you read the optional section about pointer arithmetic. Complete the function isPalindrome so that it returns true if the string cstr is a palindrome (the same backwards as forwards) and false if it is not. The function uses the cstring library.

```cpp
bool isPalindrome(char* cstr)
{
    char* front = cstr;
    char* back = cstr + strlen(cstr)-1;
    while (front < back)
    {
        // Complete code here
    }
    return true;
}
```

3.4 Assignment 4

In statistics, the mode of a set of values is the value that occurs most often or with the greatest frequency. Write a function that accepts as arguments the following:

A) An array of integers

B) An integer that indicates the number of elements in the array

The function should determine the mode of the array. That is, it should determine which value in the array occurs most often. The mode is the value the function should return. If the array has no mode (none of the values occur more than once), the function should return −1. (Assume the array will always contain nonnegative values, a[i] is from 0 to 1.000.000)

Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.

3.5 Assignment 5

In statistics, when a set of values is sorted in ascending or descending order, its median is the middle value. If the set contains an even number of values, the median is the mean, or average, of the two middle values. Write a function that accepts as arguments the following:

A) An array of integers

B) An integer that indicates the number of elements in the array

The function should determine the median of the array. This value should be returned as a double. (Assume the values in the array are already sorted.)

Demonstrate your pointer prowess by using pointer notation instead of array notation in this function.

3.6 Assignment 6

Write a function that accepts an int array and the array's size as arguments. The function should create a new array that is twice the size of the argument array. The function should copy the contents of the argument array to the new array and initialize the unused elements of the second array with 0. The function should return a pointer to the new array.

3.7 Assignment 7

One of your professors has asked you to write a program to grade her final exams, which consist of only 20 multiple-choice questions. Each question has one of four possible answers: A, B, C, or D. The file **CorrectAnswers.txt** contains the correct answers for all of the questions, with each answer written on a separate line.

- The first line contains the number of answers.
- The second line contains the answer to the first question
- The third line contains the answer to the second question, and so forth.

Write a program that reads the contents of the **CorrectAnswers.txt** file into a char array, and then reads the contents of another file, containing a student's answers, into a second char array **StudentAnswers.txt** (the format of StudentAnswers.txt is the same as CorrectAnswers.txt).

The program should determine the number of questions that the student missed and then display the following:

- A list of the questions missed by the student, showing the correct answer and the incorrect answer provided by the student for each missed question
- The total number of questions missed
- The percentage of questions answered correctly. This can be calculated as

Correctly Answered Questions / Total Number of Questions

If the percentage of correctly answered questions is 70% or greater, the program should indicate that the student passed the exam. Otherwise, it should indicate that the student failed the exam.
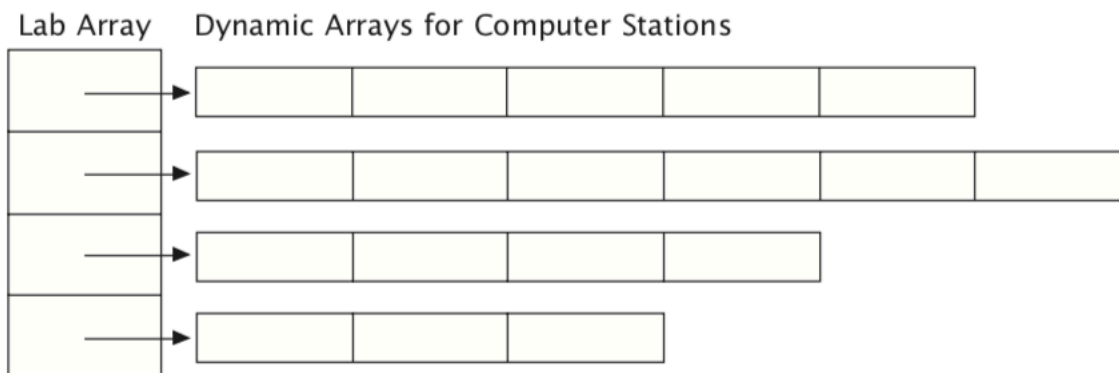
3.8 Assignment 8

You run M computer labs. each lab contains Ni computer stations.

Each user has a unique five-digit ID number. Whenever a user logs on, the user's ID, lab number, and the computer station number are transmitted to your system. For example, if user 49193 logs onto station 2 in lab 3, then your system receives 49193 2 3 as input data. Similarly, when a user logs off a station, then your system receives the lab number and computer station number.

Write a computer program that could be used to track, by lab, which user is logged onto which computer. For example, if user 49193 is logged into station 2 in lab 3 and user 99577 is logged into station 1 of lab 4, then your system might display the following:

Lab	**Station**					
	1	2	3	4	5	6
1	0	0	0	0	0	
2	0	0	0	0	0	0
3	0	49193	0	0		
4	99577	0	0			

Create a menu that allows the administrator to simulate the transmission of information by manually typing in the login or logoff data. Whenever someone logs in or out, the display should be updated. Also write a search option so that the administrator can type in a user ID and the system will output what lab and station number that user is logged into, or "None" if the user ID is not logged into any computer station.

Lab Array   Dynamic Arrays for Computer Stations

3.9 Assignment 9

A local zoo wants to keep track of how many pounds of food each of its n monkeys eats each day during a typical week. Write a program that stores this information in a two-dimensional n × 7 array, where each row represents a different monkey and each column represents a different day of the week. The program should first have the user input the data for each monkey. Then it should create a report that includes the following information:

- Average amount of food eaten per day by the whole family of monkeys.
- The least amount of food eaten during the week by any one monkey.
- The greatest amount of food eaten during the week by any one monkey.

Input Validation: Do not accept negative numbers for pounds of food eaten.

3.10 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
                1234567890123456789012345678 90
   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.

3.11 Assignment 11 – PA (Project Assignment)

1. Create a private GitHub repository.
2. Invite all group members to the repository.
3. Clone the repository to your local machines.
4. Create a folder to store the project's source code files.
5. Implement an empty main() function.
6. Add a .gitignore file to the project.
7. Develop additional functions and files to illustrate the use of Git commands: git add, git commit, git push, and git pull.
8. Showcase an instance of a code conflict and demonstrate how to resolve it.

Each student is required to compose a concise report, including some screenshots, to document their individual contributions to this assignment.