CS 122 Lab 3

Multi-dimension arrays Paul Cao

Many applications using arrays that have more than one dimensions. In this lab, we will learn process 2-dimensional arrays. The code for this lab is due before midnight on Thursday, 2/3/2011.

Pre-Lab

Read the following paragraphs about using 2-D arrays. You don't have to turn in anything for the pre-lab section.

2-dimensional arrays are defined in a similar way as the 1-D arrays. Imagine a container with two columns of partitions that one can use to store pills. Suppose that a person takes 2 pills a day. He would load the container for her every Sunday night and she was good to go for the rest of the week. She only had to know where the pills for each day were and which one of the pills was a day pill and which one was night pill. We would tell her that the left ones were day pills and the right ones were night pills. The container had 7 rows for 7 days of the week and two columns one for the day and the other for the night. If we would view the container as a 2-D array, we would define it as: `pill container[7][2]`.

Here we use two brackets; one to define the number of rows, and the other to define the number of columns. Similarly, a 2-D array of integers with 3 rows and 4 columns would be defined as: `int x[3][4];`

Now let's write a program that uses a 2-D arrays.

```
// This program will ask a runner for her/his fastest 5 times for 6
// different distances and will display them using a 2-D array.

#include<iostream>
using namespace std;

int find_distance(int j);   //a function that returns a distance based on the choice j

int main( ){
     int i, j;
     int distance[6];
     double data[6][5];  //This array will keep 30 values in 6 rows and 5 columns
     // 6 events and 5 times for each one of the events

     for(j = 0; j < 6; j++){
        distance[j] = find_distance(j);
        cout << "\nEnter 5 of your best running times for \n " <<  distance[j] << " m \n";
        for(i = 0;  i < 5; i++){
               cout << "Enter a time \n";
               cin >> data[j][i];
        }

     }

     cout << "Here is your best 5 times: ";
     for(j = 0;  j < 6; j++){
        cout << "\nDistance : " << distance[j] << " m \n";
        for(i = 0; i < 5; i++){
            cout << data[j][i] << "\t";
        }
        cout << endl;
    }

    return 0;
}

int find_distance(int j)
{
    switch (j)
    {
      case 0: // 100 meter
            return 100;
            break;
      case 1: // 150 meter
            return 150;
            break;
      case 2: // 200 meter
            return 200;
            break;
      case 3: // 400 meter
            return 400;
            break;
      case 4: // 500 meter
            return 800;
            break;
      default: // 1600 meter
            return 1600;
    }
}
```

In the above program, we can access the 3rd time of the 4th event (400 m) in:

```
data[3][2];  // note that the 3rd time is stored in column with index 2
             // and the 4th event is stored in row with index 3
```

To access the 5 times for 150 m event, we can use a for loop

```
for(i = 0; i < 5; i++)
      cout<<data[1][i];
```

In-Lab

A small airline has just purchased a computer for its new automated reservations system. The president has asked you to program the new system in C++. You are to write a program to assign seats on each flight of the airline's only plane. The plane's capacity is 12 seats. There are 4 rows with 3 seats in each row.

Your program should display the following menu of alternatives:

Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"

Selection:

If the person types f, then your program should assign a seat in the front section (rows 1-2). If the person types b, then your program should assign a seat in the back section (rows 3-4). The seat should be assigned sequentially if possible. For example, if there is a free seat in row 1, you should assign it instead of assigning a seat to the user in row 2. Your program should then print a boarding pass indicating the person's seat number (row and seat numbers) and whether it is in the front or back section of the plane.

Your program should, of course, never assign a seat that has already been assigned. When the front section is full, your program should ask the person if it is acceptable to be placed in the back section (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message:

Next flight leaves in 3 hours. Please wait for the next flight.

If the whole plane is full, your code should display that the whole flight is full followed by the message above.

Your code should save the status of each seat (assigned or free) into a file and before assigning any seat, the file will be read and check if the target seat is free. Also after assigning a seat, the corresponding status of the seat in the file should be changed. Thus each time the code is run, the status file may be changed. The status file should be set up initially so all seats are free. It is up to you to decide the format of the file but a 0/1 value for each seat should be enough to indicate the status of each seat. Inside your code, you can use a 2D array to represent the status of the seats. This array's data should be read from the status file and if the array's data is changed in the code, this change should be written back to the file for storage. To make your code easier, you can define this array as a global array.

Requirements

1. Your codes should use appropriate functions to make its structure better. Your functions should also be well commented.
2. You should thoroughly test your code for all possible scenarios.
3. You should submit your status file if you use a different format other than the one given in the lab folder.

Sample Output (After each run of the code, the status file's content was changed accordingly. The content of the file is omitted here.)

(*First time the code was run*)

```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: f
Your seat information is the following.
Section:            front
Row number:         1
Seat number:        1
Enjoy your flight! Bye!
```

(*The code was run again*)
```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: b
Your seat information is the following.
Section:            back
Row number:         3
Seat number:        1
Enjoy your flight! Bye!
```

(*The code was run again*)
```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: f
Your seat information is the following.
Section:            front
Row number:         1
Seat number:        2
Enjoy your flight! Bye!
```

(*After several runs where front seats have been repeatedly selected, all front seats were exhausted. The code was run again*)
```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: f
Your preferred front section is full. Is the back ok for you? (y for
yes and n for no)y
Your seat information is the following.
Section:            back
Row number:         3
Seat number:        2
Enjoy your flight! Bye!
```

(*The code was run again*)
```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: b
Your seat information is the following.
Section:            back
Row number:         3
Seat number:        3
Enjoy your flight! Bye!
```

(*The code was run again*)
```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: f
Your preferred front section is full. Is the back ok for you? (y for
yes and n for no)n
Next flight leaves in 3 hours. Please wait for the next flight. Bye!
```

(*After several runs again where back seats have also been repeated selected, all seats were exhausted, i.e. the plane is full already. The code is run again*)

```
Welcome to C++ flight seat assignment interface!
Please type f for "front"
Please type b for "back"
Selection: b
Sorry, the whole plane is full. Next flight leaves in 3 hours. Please
wait for the next flight. Bye!
```