CSC 150

PRE LAB #11 - Multidimensional Arrays and Files

Purpose:

1. To introduce and work with 2D and multidimensional arrays
2. Work with input and output files

1. Complete the section exercises in chapter 10 of the course's textbook.

Section 10.10 exercises (pages 320 - 322)

1)
   a. true
   b. true
   c. true
   d. true
2) a
3) b
4) char 1_name[12][12];
5) double sales[31][10]
6) 210
7) char names[11][16];
   names[0] = {"Sue"};
   names[1] = {"Marie"};
8) for (i = 0; i < 11; i++)
     cout << names[i][];
9) int scores[5][30] = {0};
10) scores[2][1] = {91};
    scores[4][3] = {88};
11)
    a.
      i. for(i = 0; i < 20; i++)
           total_sisters += siblings[i][0];
     ii. for(i = 0; i < 20; i++)
           total_brothers += siblings[i][1];
    b. Foo(siblings)
    c. void Foo(int siblings[20][2])
    d. for(i = 0; i < 20; i++)
         for(j = 0; j<2; j++)
           cout << siblings[i][j];
         cout << endl;

2. Given the following code, in which cell of the array are the following values found?

```
int arr_2d [3][5];

arr_2d[1][4] = 88;
arr_2d[0][3] = 99;
```

j 88
1 99

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o

3. Given the following code, show the array contents

```
int arr_2d [3][3];

for( int i = 0; i < 3; i++ )
    for( int j = 0; j < 3; j++ )
        arr_2d[i][j] = i + j;
```

0	1	2
1	2	3
2	3	4

4. Write the code to declare an array of 4 rows by 6 columns. Using loops, fill it row-wise with the powers of 2, starting at 2^0. Do not use the pow( ) function.

```
#include <iostream>
#include <cstring>
using namespace std;


int arry[4][6];
int i = 0;
int j = 0;
int result = 1;

for(i = 0; i < 4; i++)
    for (j = 0; j < 6; j++)
        result = 2 * result
```

5. Given the code below, write the function find_max( ) that finds the **location** of the array element holding the largest value. (You can find this code in file PL-11-1.cpp)

```
// Program: PL-12-1.cpp
#include <iostream>
#include <ctime>
#include <cstdlib>

using namespace std;

const int PAGES = 5;
const int ROWS = 6;
const int COLUMNS = 7;

//your function prototype here

void fill_array( int p[ ][ROWS][COLUMNS], int pages );

int main()
{
    int data[PAGES][ROWS][COLUMNS] = { 0 } ;
    int page_ind = 0;
    int row_ind = 0;
    int col_ind = 0;

    fill_array( data, PAGES );

    find_max (   /* fill in arguments */ );  //find max in just page 0
    cout << "The maximum value in the first page is: "
         << data[page_ind][row_ind][col_ind] << endl;

    find_max (  /* fill in arguments */  ); //find max in whole array
    cout << "The maximum value in the entire array is: "
         << data[page_ind][row_ind][col_ind] << endl;

    cout << endl;
    return 0;
}

void fill_array( int d[ ][ROWS][COLUMNS], int pages )
{
    srand( unsigned ( time( NULL ) ) );

    int pg, rw, cl;

    for( pg = 0; pg < pages; pg++ )
        for( rw = 0; rw < ROWS; rw++ )
            for( cl = 0; cl < COLUMNS; cl++ )
                d[pg][rw][cl] = rand( ) % 1000;

}


//your function definition here
```

6. Complete the section exercises in chapter 11 of the course's textbook.

Section 11.2 exercises (page 352)

1) c
2) a
3)
   a. false
   b. true
   c. true
   d. true
   e. false

7. Answer the following True/False questions:

a) T F The include file <iostream> contains all the declarations necessary for file operations.

b) T F The statement below opens a file named `scores.txt` using file stream `fin`.

```
fin.open("scores.txt");
```

c) T F The operating system may limit a user's access to some files.

d) T F To close a file `values.txt` with associated with file stream `inFile`, you use the statement:

```
inFile.close("values.txt");
```

e) T F The stream extraction operator (`>>`) not only reads user input from the `cin` object, but it can also read data from a file.

f) T F In C++, you can open the same file for input and output.

g) T F When you read a value from a file with the stream extraction operator, the operator returns a value that can be evaluated as true or false.

h) T F To read to the end of a file with file stream `fin`, you can use the expression

```
while( fin >> var )
```

i) T F Values must be separated by whitespaces when the >> operator extracts data from a file.

j) T F After using the `fout.open( filename )` command for a file, the next operation you normally use is:

```
if(!fout)
```

k) T F The following statement will read a full line of text, including whitespace, storing to a cString variable.

```
fin.getline( full_name, 128 );
```

10. Modify the following program so that the data is read from the file **sales.txt** rather than from the keyboard. The computed total sales for each salesperson with his or her name should also be saved in the file **outSales.txt** in the format shown below. If either the input or output file fails to open, exit the program after displaying an error message. The program code is saved as PL-11-2.cpp.

```
// This program reads in the names of salesmen and their sales figures
// (in thousands) on the four quarters of a year. The program then
// computes the total sales for each salesmen and prints the name of a
// salesmen and his or her total sales. The program terminates when
// the user types Crtl-Z.  The data is read as: John 20 80 95 45

#include <iostream>
using namespace std;

int main()
{
   char name[40];    // Store a salesperson's name
   int sales;        // Store a salesperson's quarterly sales
   int total = 0;    // Accumulate sales

  // Read in first name
  cout << "Enter name and the four quarters sales: ";

  // Read in each name and sales
  while( cin >> name )
  {
        total = 0;
        for (int i=0; i < 4; i++)
        {
              cin >> sales;
              total += sales;
        }

        cout << name << " has total sales: " << total << endl;
        cout << "Enter name and the four quarters sales: ";

  }

  return 0;
}
```

Sample input file content:

```
Sally 23 92 86 105
Ed 45 34 76 23
Joe 76 45 76 76
```

Sample output file content:

```
Sally 306
Ed 178
Joe 273
```