Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms

Summer 2020 - 2021

Lab Guide #3 – Week 2 – 1

OBJECTIVE: Two-Dimensional Arrays with Pointers and Dynamic Memory Allocation

Instructor : Okyay SAY
Assistant : Ruşen ASAN

Q1. Suppose that a C program contains the following statement:

```
double matrix[4][3] = { { 34.25, 21.42, 65.78 },{ 81.52, 21.95, 20.12 },{ 21.96, 10.3, 50.17 },{ 40.47, 59.84, 35.26 } };
```

According to the statement, examine the values of:

- matrix
- (matrix + 1)
- *(matrix + 1)
- (*(matrix) + 1)
- *(*(matrix) + 1)
- *(*(matrix + 1))
- *(*(matrix) + 1) + 2
- (*(matrix + 1) + 2)
- *(*(matrix + 1) + 2)

Check the results by the program.

Project Name: LG3_Q1
File Name: Q1.cpp

Q2. Write a C program that will initialize a two-dimensional integer array of size 3 by 3 with the numbers 3, 9, 27 ... in a loop. After storing these values, the program should also give an output of these values to the text file result.txt in the given format below, including their addresses and pointer iterations.

Example Run:

```
Successful! Please see the result.txt file for the output.
```

Content of the result.txt file

```
Element Name                 Value   Address
-----------------------      -----   --------
*(*(nums +   0) +   0)           3   00EFFA9C
*(*(nums +   0) +   1)           9   00EFFAA0
*(*(nums +   0) +   2)          27   00EFFAA4
*(*(nums +   1) +   0)          81   00EFFAA8
*(*(nums +   1) +   1)         243   00EFFAAC
*(*(nums +   1) +   2)         729   00EFFAB0
*(*(nums +   2) +   0)        2187   00EFFAB4
*(*(nums +   2) +   1)        6561   00EFFAB8
*(*(nums +   2) +   2)       19683   00EFFABC
```

Project Name: LG3_Q2
File Name: Q2.cpp

Q3. Write a C program that gets the number of data and integer values from the user and creates an integer array **dynamically** to keep them. It will continue until a non-positive number is given. The program will calculate and display the products of these numbers for each time.

Write the following functions;

- **input :** inputs a given number of integer values into a dynamic memory area whose address is received as a pointer.
- **findPro :** receives the address of a dynamic integer memory area together with size, and returns the products of those values.

Example Run:

```
How many numbers will be entered? 4
Enter 4 numbers: 15 12 3 6
The products of the numbers: 3240.00

How many numbers will be entered? 2
Enter 2 numbers: 15 58
The products of the numbers: 870.00

How many numbers will be entered? -1
```

Project Name: LG3_Q3
File Name: Q3.cpp

The %s operator is used for reading strings of characters into character arrays using the scanf function.

Q4. Write a C program that reads words from "**words.txt"**. It reads the size of file and words into a **dynamically** created **two-dimensional array.** After reading, write all of the words to the screen as in the example run. The maximum number of characters in a word is **10**. (The first line of the file consists the number of words.)

Write the following functions;

- **readFile:** gets a file pointer and a two-dim character array, to read each word line by line and store them into a two-dim character array.
- **displayWords:** gets a one-dim character array and displays the words in this array.
- **displayWordSpecCol:** gets a two-dim array and row size as parameters. Then, it gets a column number from the user and displays the word in that column.

Example Run:

```
Enter the number of words in the text file: 6

WORDS:
***********
jzfhab
ooromi
ymecaz
fbekzo
uizeon
leeyne

Enter the column number: 3
Freeze
```

words.txt

```
jzfhab
ooromi
ymecaz
fbekzo
uizeon
leeyne
```

Project Name: LG3_Q4
File Name: Q4.cpp

Additional Questions

AQ1. A statistical research company wants to analyze the number of **premature, normal weight, overweight** newborn babies in a week at 5 hospitals shown at the below table.

	premature	normal weight	overweight
Bayındır	6	45	5
Zekâi Tahir Burak	15	96	2
TOBB	34	32	8
Güven	8	47	9
Hacettepe	17	65	3

Write a C Program that reads the .txt file in to a two-dimensional array and finds the **sum** of **each column** and **row** by using the below functions and **display** the result.

Write the following functions:

- **read:** gets a file pointer and a two-dim array to read the data from the text file named as **"hospital.txt"** and store them into a two-dim array.
- **findTotals:** finds the **total** number of the newborn babies in every hospital and the **total** number of the newborn babies in every group (premature, normal weight, overweight) **and** stores them into different one-dimensional arrays.
- **sum:** takes a one-dimensional integer array as a parameter. Then the function finds and returns the **sum** of the data in the one-dimensional array.
- **display:** takes a two-dimensional array and two one-dimensional arrays as parameters and it displays the content of the arrays as shown in the example run. Also find and display the **total number** of babies born in a week at all hospitals using **sum** function.

Example Run:

```
       6 +    45 +     5    =      56
      15 +    96 +     2    =     113
      34 +    32 +     8    =      74
       8 +    47 +     9    =      64
      17 +    65 +     3    =      85
+_____________________     +    _____

      80 +   285 +    27    =     392
```

Project Name: LG3_AQ1
File Name: AQ1.cpp

AQ2. Write a C program that reads the ids of some cities and the number of tourists that came to these cities during 5 years from a text file named "**tourism.txt**". The content of the text file is shown below.

- The first row of the file contains the years to be stored in a one-dimensional array.
- The first column of the file contains the Ids of the cities to be stored in a one-dimensional array.
- The rest of the file contains the number of tourists to be stored in a two-dimensional array.

tourism.txt

```
      2001    2002    2003    2004    2005
06    22000   56000   43000   70000   12000
42    44000   27000   10000   32000   80000
27    21000   89000   16500   65000   14000
07   130000   33000   30000   20000   62000
35    67000   54000   19000   17000   43000
50    90000   61200   13000   52000   74000
34    23000   76000   72000   56000   49000
77    80000   10000   11600   13000   79000
17    43000   45000   15000   58000   70600
33    65000   54000   18100   41000   10100
26    20000   40000   15200   23000   40000
```

The program should find and display

- for each city:
  - the year that the maximum number of tourists came,
  - the total number of tourists came.
- for each year:
  - the cities that the minimum number of tourists came,
  - the total number of tourists came.

Write the following functions:

- **colMaxVal:** finds and returns the column index of the maximum value in a certain row of a 5-column two-dimensional array.
- **rowMinVal:** finds and returns the row index of the minimum value in a certain column of a 5-column two-dimensional array.
- **sumRow:** finds the sum of each row of a 5-column two-dimensional array and stores them in a one-dimensional array.
- **sumCol**: finds the sum of each column of a 5-column two-dimensional array and stores them in a one-dimensional array.

Example Run:

```
City Id        Year with Max. Tourist     Total Tourists
**********     **********************     **************
 6                      2004                   223000
42                      2005                   203000
27                      2002                   205500
 7                      2001                   275000
35                      2001                   200000
50                      2001                   270200
34                      2002                   276000
77                      2001                   193600
17                      2005                   242000
33                      2002                   158200
26                      2002                   151200

Year     City with Min. Tourist        Total Tourists
****     ************************      **************
2001                      26               585000
2002                      77               538200
2003                      42               273400
2004                      77               457400
2005                      33               543700
```

Project Name: LG3_AQ2
File Name: AQ2.cpp