

## TASK 1

First create a new folder "Lab4\_YourLastName"

**mkdir Lab4\_YourLastName**

Ex. Lab4\_Pare

Once this folder has been created, we will want to enter it, now we are still in the home directory.

**cd Lab4\_YourLastName**

First you should download the .tar file and extract the files into the folder you have been working with so far.

**tar -xvf Lab4\_provided\_code.tar.gz**

In this lab, the makefile is already provided. Make sure you understand how this makefile is different than the one you created for the c programs

<b>g++</b>	is the compiler used for c++
<b>-Wall</b>	this flag is used to turn on most compiler warnings
<b>-g</b>	this flag adds debugging information to the executable file
<b>-std=c++11</b>	as of GCC 4.8.1, GCC's C++11 mode implements all of the major features of C++11

## TASK 2

This lab explores different operations to be performed on matrices. Some of the code is already provided. You should implement the remaining functions to complete the lab.

### First compile the provided code by using make

The terminal should now have created an executable called “prog” you can run this like any other Executable

**./prog**

You will be provided with the following options

1. Print the Matrix
  2. Perform an addition of the matrix with itself
  3. Get the maximum value in the matrix
  4. Find whether an element exists in the matrix
  5. Change an existing element in the matrix
  6. Perform matrix multiplication
  7. Transpose of a matrix
- Which operations would you like to perform on the matrix: 7

**Enter 1**

This will print the Matrix that the function “**createMatrix()**” reads in from the file “input.txt”

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

**Enter 2**

This will perform the addition of the matrix with itself using the “**addMatrix()**” function. You will see the following results

2	4	6	8
10	12	14	16
18	20	22	24
26	28	30	32

Next, implement the functionality of number 3 – Get the maximum value in the matrix.

See how other functions in the “matrix.cpp” and “matrix.h” are defined and code the “**getMax()**” function. Use the “grid” matrix as the base matrix.

After implemented, the following should be printed in the terminal

**Which operations would you like to perform on the matrix: 3**

**16**

Next, implement the functionality of number 4 – Find whether an element exists in the matrix.

See how other functions in the “matrix.cpp” and “matrix.h” are defined and code the “**findElement(int no)**” function. Use the “grid” matrix as the base matrix.

After implemented, the following should be printed in the terminal

**Which operations would you like to perform on the matrix: 4**

**Enter the number to find: 6**

**Element found at 1, 1**

Next, implement the functionality of number 5 – Change the element in matrix.

This will ask the user to provide a **valid** row and column number of an element in the matrix and replace it with the provided number.

See how other functions in the “matrix.cpp” and “matrix.h” are defined and code the “**changeElement(int r, int c, int no)**” function.

After implemented, the following should be printed in the terminal

**Enter the row of the element to change: 1**

**Enter the column of the element to change: 0**

**Enter the value to replace: 99**

1	2	3	4
99	6	7	8
9	10	11	12
13	14	15	16

Next, implement the functionality of number 6 – Perform matrix multiplication

See how other functions in the “matrix.cpp” and “matrix.h” are defined and code the “**multiplyMatrix()**” function. Use the “grid” matrix as the base matrix and store the multiplication result in “grid\_temp”. Print the contents of “grid\_temp” once done.

Refer to the “**addMatrix()**” function

After implemented, the following should be printed in the terminal

**Which operations would you like to perform on the matrix: 6**

90	100	110	120
202	228	254	280
314	356	398	440
426	484	542	600

Next, implement the functionality of number 7 – Transpose of a matrix

See how other functions in the “matrix.cpp” and “matrix.h” are defined and code the “**transposeMatrix()**” function. Use the “grid” matrix as the base matrix and store the multiplication result in “grid\_temp”. Print the contents of “grid\_temp” once done.

After implemented, the following should be printed in the terminal

**Which operations would you like to perform on the matrix: 7**

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

### TASK 3

From your home directory:

**tar -zcvf Lab4\_YourLastName.tar.gz Lab4\_YourLastName**

A tar file should be created, and this is what you will be turning into canvas for grading. If you want to decompress your tar file and check its contents you can

**tar -zxvf Lab4\_YourLastName.tar.gz**

Do not forget to check the grading rubric.