Course: Programming Fundamentals – **ENCM 339**

Lab #: Lab 1

Instructor: S. Norman

Student Name: **Mitchell Sawatzky**

Lab Section: **B02**

Date Submitted: **Sept 15, 2015Exercise E**

brit-leap.c

// ENCM 339 Fall 2015 Lab 1 Exercise C

#include <stdio.h>

int is\_leap\_in\_GB(int year);

// Returns 1 if year is leap in Britain, and 0 otherwise.

// (Return value doesn't necessarily sense if year is earlier than

// the adoption of AD year numbering in Britain!)

void call\_is\_leap(int year);

// Prints year embedded in a message about whether year was leap in Britain.

int main(void)

{

call\_is\_leap(1700);

call\_is\_leap(1701);

call\_is\_leap(1704);

call\_is\_leap(1800);

call\_is\_leap(1812);

call\_is\_leap(1815);

call\_is\_leap(2000);

call\_is\_leap(2015);

call\_is\_leap(2016);

call\_is\_leap(2100);

call\_is\_leap(2102);

call\_is\_leap(2196);

return 0;

}

int is\_leap\_in\_GB(int year)

{

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)

return 1;

else

return 0;

}

void call\_is\_leap(int year)

{

printf("The year %d was, is, or will be ", year);

if (!is\_leap\_in\_GB(year))

printf("non-");

printf("leap in Britain.\n");

}

Terminal Output

Mitchell@ttys003 14:34 {126} [lab1]$ ./test.out

The year 1700 was, is, or will be non-leap in Britain.

The year 1701 was, is, or will be non-leap in Britain.

The year 1704 was, is, or will be leap in Britain.

The year 1800 was, is, or will be non-leap in Britain.

The year 1812 was, is, or will be leap in Britain.

The year 1815 was, is, or will be non-leap in Britain.

The year 2000 was, is, or will be leap in Britain.

The year 2015 was, is, or will be non-leap in Britain.

The year 2016 was, is, or will be leap in Britain.

The year 2100 was, is, or will be non-leap in Britain.

The year 2102 was, is, or will be non-leap in Britain.

The year 2196 was, is, or will be leap in Britain.

**Exercise F**

Lab1exF.c

// ENCM 369 Fall 2015 Lab 1 Exercise F

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int max;

int scan\_count;

int column;

int row;

printf("How many rows would you like in your times table?\n");

printf("Enter a number between 1 and 15.\n");

scan\_count = scanf("%d", &max);

// If scanf couldn't read an int at all, print an error message and quit.

if (scan\_count == 0)

{

printf("Error: Invalid Input");

exit(1);

}

// If scanf got an out-of-range number, print the number within

// an error message and quit.

if ((1 > max) || (max > 15))

{

printf("Error: %d is out of range!", max);

exit(1);

}

printf("Here is your table ...\n\n");

// Print the header row of the table.

// Note that %4d prints an int using a \*minimum\* of 4 characters,

// which will help to get input lined up in columns.

printf(" X");

for (column = 1; column <= max; column++)

printf("%4d", column);

printf("\n");

// Print all the other rows of the table, using a nested loop.

for (row = 1; row <= max; row++)

{

printf(" %d", row);

for (column = 1; column <= max; column++)

printf("%4d", row \* column);

printf("\n");

}

return 0;

}

Terminal Output

Mitchell@ttys003 14:42 {0} [lab1]$ ./test.out

How many rows would you like in your times table?

Enter a number between 1 and 15.

3

Here is your table ...

X 1 2 3

1 1 2 3

2 2 4 6

3 3 6 9

Mitchell@ttys003 14:43 {0} [lab1]$ ./test.out

How many rows would you like in your times table?

Enter a number between 1 and 15.

5

Here is your table ...

X 1 2 3 4 5

1 1 2 3 4 5

2 2 4 6 8 10

3 3 6 9 12 15

4 4 8 12 16 20

5 5 10 15 20 25

Mitchell@ttys003 14:43 {0} [lab1]$ ./test.out

How many rows would you like in your times table?

Enter a number between 1 and 15.

6

Here is your table ...

X 1 2 3 4 5 6

1 1 2 3 4 5 6

2 2 4 6 8 10 12

3 3 6 9 12 15 18

4 4 8 12 16 20 24

5 5 10 15 20 25 30

6 6 12 18 24 30 36

Mitchell@ttys003 14:43 {0} [lab1]$