**Date Submitted: 10/9/18**

**------------------------------------------------------------------------------------**

**Task 01:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> // enables us to use the sinf()

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/fpu.h" // enables us to use Floating Point Units

**#include** "driverlib/sysctl.h"

**#include** "driverlib/rom.h"

//

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

**#define** SERIES\_LENGTH 100 // the depth of our data

**float** gSeriesData[SERIES\_LENGTH]; // an array of floats

int32\_t i32DataCount = 0; // counter for math loop

**int** **main**(**void**)

{

**float** fRadians; // calculate sine

ROM\_FPULazyStackingEnable();

ROM\_FPUEnable(); // Enable Floating Points

// 50MHz System Clock

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // getting a full sine wave

// Calculate the sine value for each angle

**while**(i32DataCount < SERIES\_LENGTH)

{

gSeriesData[i32DataCount] = **sinf**(fRadians \* i32DataCount);

i32DataCount++;

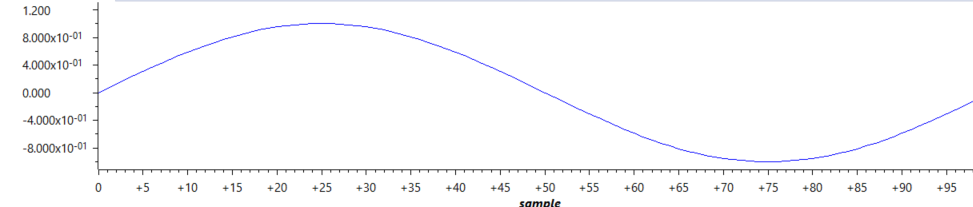
}

**while**(1)

{

}

}



**------------------------------------------------------------------------------------**

**Task 02:**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h> // enables us to use the sinf()

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/fpu.h" // enables us to use Floating Point Units

**#include** "driverlib/sysctl.h"

**#include** "driverlib/rom.h"

//

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

**#define** SERIES\_LENGTH 100 // the depth of our data

**float** gSeriesData[SERIES\_LENGTH]; // an array of floats

int32\_t i32DataCount = 0; // counter for math loop

**int** **main**(**void**)

{

**float** fRadians; // calculate sine

**float** sine;

**float** cosine;

ROM\_FPULazyStackingEnable();

ROM\_FPUEnable(); // Enable Floating Points

// 50MHz System Clock

ROM\_SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); // getting a full sine wave

// Calculate the sine value for each angle

**while**(i32DataCount < SERIES\_LENGTH)

{

sine = **sinf**(fRadians \* 50 \* i32DataCount);

cosine = **cosf**(fRadians \* 200 \* i32DataCount);

gSeriesData[i32DataCount] = 1.5 + sine + cosine;

i32DataCount++;

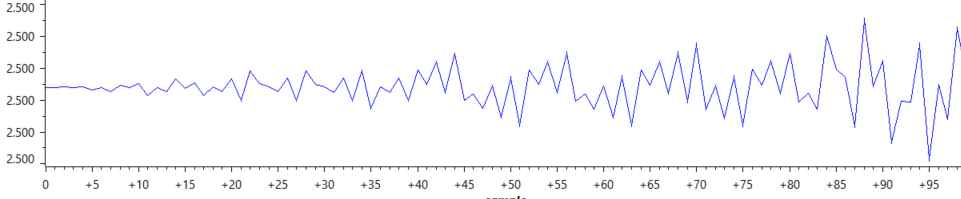
}

**while**(1)

{

}

}



**------------------------------------------------------------------------------------**