**Date Submitted: 10/30**

**Task 01:**

Youtube Link: [**https://youtu.be/90Y9LqBM5Cg**](https://youtu.be/90Y9LqBM5Cg)

**Modified Code:**

**Comment the given code.**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h>

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

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

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

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

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

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846 // Pi

**#endif**

// Depth of the buffer

**#define** SERIES\_LENGTH 100

// Buffer to store the computed floating-point values

**float** gSeriesData[SERIES\_LENGTH];

int32\_t i32DataCount = 0;

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

{

// Radian value

**float** fRadians;

ROM\_FPULazyStackingEnable(); // Enables lazy stacking of floating-point registers

ROM\_FPUEnable(); // Enables the FPU since it's disabled by default

// System clock set to 50 MHz

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

// Formula to compute and scale the radian value

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH);

// Computes the sine wave

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

{

// Computes the sine wave using the sine function and stores it in the buffer

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

// Update the index to compute the next sine point

i32DataCount++;

}

**while**(1)

{

}

}

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

**Task 02:**

Youtube Link: [**https://youtu.be/4N5YZsBvtCA**](https://youtu.be/4N5YZsBvtCA)

**Modified Code:**

**Change the original function to the specified one and increase the SERIES\_LENGTH to 1000.**

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** <math.h>

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

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

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

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

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

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846 // Pi

**#endif**

// Depth of the buffer

**#define** SERIES\_LENGTH 1000

// Buffer to store the computed floating-point values

**float** gSeriesData[SERIES\_LENGTH];

int32\_t i32DataCount = 0;

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

{

// Radian value

**float** fRadians;

ROM\_FPULazyStackingEnable(); // Enables lazy stacking of floating-point registers

ROM\_FPUEnable(); // Enables the FPU since it's disabled by default

// System clock set to 50 MHz

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

// Formula to compute and scale the radian value

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH);

// Computes the sine wave

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

{

// Computes the given function

gSeriesData[i32DataCount] = **sinf**(fRadians \* 50 \* i32DataCount) +

0.5 \* **cosf**(fRadians \* 200 \* i32DataCount);

// Update the index to compute the next sine point

i32DataCount++;

}

**while**(1)

{

}

}

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