**Task 01: Submit a comprehensive commented file of the original code**

**#include** <stdint.h> //variable definitions for the C99

**#include** <stdbool.h> //boolean definitions for the C99

**#include** <math.h> //code uses the sinf() function prototyped

**#include** "inc/hw\_memmap.h" //macros defining the memory map

**#include** "inc/hw\_types.h" //defines common types and macros

**#include** "driverlib/fpu.h" //support for Floating Point Unit

**#include** "driverlib/sysctl.h" //defines and macros for System Control API

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

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

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

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

int32\_t i32DataCount = 0; //a counter for our computation loop

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

{

**float** fRadians; //float variable to calculate sine

**FPULazyStackingEnable**(); //turn on Lazy Stacking

**FPUEnable**(); //turn on FPU

//set up system clock for 50MHz

**SysCtlClockSet**(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); //divide 2pi by the depth of array

//while loop to calculate the sine value for each of the 100 values of the angle

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

{

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

i32DataCount++;

}

//endless loop

**while**(1)

{

}

}

**Task 02: Modify the code to implement the below equation with 1000 sample points and determine the time for FPU operation.**

**#include** <stdint.h> //variable definitions for the C99

**#include** <stdbool.h> //boolean definitions for the C99

**#include** <math.h> //code uses the sinf() function prototyped

**#include** "inc/hw\_memmap.h" //macros defining the memory map

**#include** "inc/hw\_types.h" //defines common types and macros

**#include** "driverlib/fpu.h" //support for Floating Point Unit

**#include** "driverlib/sysctl.h" //defines and macros for System Control API

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

**#ifndef** M\_PI

**#define** M\_PI 3.14159265358979323846

**#endif**

**#define** SERIES\_LENGTH 1000 //the depth of data buffer to hold values is 1000

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

int32\_t i32DataCount = 0; //a counter for our computation loop

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

{

**FPULazyStackingEnable**(); //turn on Lazy Stacking

**FPUEnable**(); //turn on FPU

//set up system clock for 50MHz

**SysCtlClockSet**(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN);

//while loop to calculate the sine value for each of the 100 values of the angle

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

{

gSeriesData[i32DataCount] = (-0.06366) + (0.12613\*(**cosf**(M\_PI\*i32DataCount/500)))+(0.12258\*(**cosf**(M\_PI\*i32DataCount/250)))

+(0.01593\*(**sinf**(M\_PI\*i32DataCount/500)))+(0.03147\*(**sinf**(M\_PI\*i32DataCount/250)));

i32DataCount++;

}

//endless loop

**while**(1)

{

}

} 