

(Highlighting: Task 01(no highlighting), Task 02)

Task01: Submit a comprehensive commented file of the original code

```
/*Shabrya Lott
 * Tiva_c Lab09
 * Usage: This is a simple program enables the FPU to run and profile floating-point code
 * INPUT: NONE
 * OUTPUT: Graph
 */
#include <stdint.h>           //Variable definitions for the C99 standard
#include <stdbool.h>          //Boolean definitions for the C99 standard
#include <math.h>              //supports the sinf() function prototyped by this header file
#include "inc/hw_memmap.h"    // Macros defining the memory map of the device
#include "inc/hw_types.h"     //defines macros defining the memory map of the device
#include "driverlib/fpu.h"    //supports the Floating Point Unit
#include "driverlib/sysctl.h" //Prototypes for the system control driver
#include "driverlib/rom.h"    //provides the reg used for the tivaC

#ifndef M_PI                  //Defines pie if not defined
#define M_PI                  3.14159265358979323846
#endif

#define SERIES_LENGTH 100    //defines and set length to 100
float gSeriesData[SERIES_LENGTH]; //creates an array of floats with length of 100

int32_t i32DataCount = 0;    //counter for the computation loop

int main(void)
{
    float fRadians;          //float value used to store sine wave calc

    FPU_LazyStackingEnable(); //Allows floating point values to be used inside interrupt handler
    FPU_Enable();              //enables the floating point unit

    //sets clock to 50MHz
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    //calculates a full sine wave cycle
    fRadians = ((2 * M_PI) / SERIES_LENGTH);

    //calculate the sine value for each of the 100 values of the angle and place in array
    while(i32DataCount < SERIES_LENGTH)
    {
        //place the calculated value into the array
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);

        //increment counter
        i32DataCount++;
    }

    while(1)                //infinite loop
    {
    }
}
```

Task02: Modify the code to human heartbeat signal with 1000 sample points and determine the time for the FPU operation.

```
#include <stdint.h>           //Variable definitions for the C99 standard
#include <stdbool.h>          //Boolean definitions for the C99 standard
#include <math.h>              //supports the sinf() function prototyped by this header file
#include "inc/hw_memmap.h"    // Macros defining the memory map of the device
#include "inc/hw_types.h"     //defines macros defining the memory map of the device
#include "driverlib/fpu.h"    //supports the Floating Point Unit
#include "driverlib/sysctl.h" //Prototypes for the system control driver
#include "driverlib/rom.h"    //provides the reg used for the tivaC

#ifndef M_PI                  //Defines pie if not defined
#define M_PI                  3.14159265358979323846
#endif

#define SERIES_LENGTH 1000    //defines and set length to 1000

float gSeriesData[SERIES_LENGTH]; //creates an array of floats with length of 100

int32_t i32DataCount = 0;      //counter for the computation loop

int main(void)
{
    FPULazyStackingEnable();    //Allows floating point values to be used inside interrupt handler
    FPUEnable();                //enables the floating point unit

    //sets clock to 50MHz
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    //calculate the sine value for each of the 1000 values of the angle and place in array
    while(i32DataCount < SERIES_LENGTH)
    {
        //place the calculated value into the array
        //Equation found at:
        //https://www.reddit.com/r/Physics/comments/30royq/whats_the_equation_of_a_human_heart_beat/
        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)));
        //increment counter
        i32DataCount++;
    }

    while(1)                  //infinite loop
    {
    }
}
```