Joseph Sharp Halpin
CpE 403 Section 1001
10/16/2018

Task 01:
Youtube Link: https://youtu.be/cUOoxPiHxD0

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"

//hardcode Pi value
#ifndef M_PI
#define M_PI                    3.14159265358979323846
#endif

//define amount of data point for wave
#define SERIES_LENGTH 100

float gSeriesData[SERIES_LENGTH];

int32_t i32DataCount = 0;

int main(void)
{
    float fRadians;

    //enable Lazy Stack to reduce latency
    FPULazyStackingEnable();
    //enable FPU
    FPUEnable();

    //set clock rate for board
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);

    //calculate radians used for waveform
    fRadians = ((2 * M_PI) / SERIES_LENGTH);

    //do while i32DataCount is less than SERIES_LENGTH
    while(i32DataCount < SERIES_LENGTH)
    {
        //get the waveform value
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);

        //increment i32DataCount
        i32DataCount++;
    }

    while(1)
    {
    }
}
```

Task 02:
Youtube Link: https://youtu.be/Gy52OiOuA1I

Code:

```c
#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"

//hardcode Pi value
#ifndef M_PI
#define M_PI                    3.14159265358979323846
#endif

//define amount of data point for wave
#define SERIES_LENGTH 1000

float gSeriesData[SERIES_LENGTH];

int32_t i32DataCount = 0;

int main(void)
{
    float fRadians;

    //enable Lazy Stack to reduce latency
    FPULazyStackingEnable();
    //enable FPU
    FPUEnable();

    //set clock rate for board
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ | SYSCTL_OSC_MAIN);



    //calculate radians used for waveform
    fRadians = ((2 * M_PI) / SERIES_LENGTH);

    //do while i32DataCount is less than SERIES_LENGTH
    while(i32DataCount < SERIES_LENGTH)
    {
        //get the waveform value
        gSeriesData[i32DataCount] = 1.5 + 1.0*sinf(fRadians * 50 * i32DataCount) + 0.5*cosf(fRadians * 200 * i32DataCount);

        //increment i32DataCount
        i32DataCount++;
    }

    while(1)
    {
    }
}
```