Brian West
Github root directory: https://github.com/westbrian2/Fall2019

**Date Submitted: 11/12/19**

**Task 00: Execute provided code**

```
#define TARGET_IS_BLIZZARD_RB1
#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"
#include "driverlib/rom_map.h"

#ifndef M_PI
#define M_PI                    3.14159265358979323846
#endif

#define SERIES_LENGTH 100

float gSeriesData[SERIES_LENGTH];

int32_t i32DataCount = 0;

int main(void)
{
    float fRadians;

    ROM_FPULazyStackingEnable();
    ROM_FPUEnable();

    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    fRadians = ((2 * M_PI) / SERIES_LENGTH); // determine value

    while(i32DataCount < SERIES_LENGTH) // number of data points is under
series length
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount); assign
value to array

        i32DataCount++;
    }

    while(1)
    {
    }
}
```

**Grading scheme:** 30% Coding, 30% Documentation, 40% Execution/Video.

Brian West
Github root directory: https://github.com/westbrian2/Fall2019

---

## Task 01:

**Modified Code:**

```
#define TARGET_IS_BLIZZARD_RB1
#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"
#include "driverlib/rom_map.h"

#ifndef M_PI
#define M_PI                    3.14159265358979323846
#endif

#define SERIES_LENGTH 100

float gSeriesData[SERIES_LENGTH];

int32_t i32DataCount = 0;

int main(void)
{
    float fRadians;

    ROM_FPULazyStackingEnable();
    ROM_FPUEnable();

    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);

    fRadians = ((2 * M_PI) / SERIES_LENGTH); // determine value

    while(i32DataCount < SERIES_LENGTH) // number of data points is under
series length
    {
        gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount); assign
value to array

        i32DataCount++;
    }
```

```
    while(1)
    {
    }
}
-------------------------------------------------------------------------------
------
```

**Grading scheme:** 30% Coding, 30% Documentation, 40% Execution/Video.