1. Write a function in ARM Cortex-M4 assembly language to calculate the area of a circle. Write a C program to test your function. The function prototype is:

```
float  CircleArea(float radius) ;
```

```
CircleArea: // S0 = radius
            VMUL.F32   S0,S0,S0   // S0 = radius * radius
            VLDR       S1,pi      // S1 = 3.14159
            VMUL.F32   S0,S0,S1   // S0 = 3.14159*radius*radius
            BX         LR

pi:         .float     3.14159
```

2. Write a function in ARM Cortex-M4 assembly language to compute the dot product of two vectors. Write a C program to test your function. The function prototype is:

```
float DotProduct(float vec1[], float vec2[], int32_t len) ;
```

```
DotProduct: // R0 = &vec1[0], R1 = &vec2[0], R2 = len
            VMOV       S0,0.0     // sum = 0.0
L1:         CBZ        R2,L2      // Done if len == 0
            VLDR       S1,[R0]    // S1 = *vec1
            ADD        R0,R0,4    // vec1++
            VLDR       S2,[R1]    // S2 = *vec2
            ADD        R1,R1,4    // vec2++
            VMLA.F32   S0,S1,S2   // sum += S1*S2
            SUB        R2,R2,1    // items--
            B          L1         // repeat
L2:         BX         LR         // return
```

**Replace these 4 by:**

```
VLDMIA  R0!,{S1}
VLDMIA  R1!,{S2}
```

3. Write a function in ARM Cortex-M4 assembly language to evaluate a polynomial. Write a C program to test your function. The function prototype is:

```
float  Polynomial(float x, float coef[], int32_t terms) ;
```

This is your lab assignment!

4. Write a function in C that calls the Polynomial function developed in problem 3 to compute an eight term Taylor series approximation to the inverse, $X^{-1}$. (Do not use a divide!) Write a C program to test your function. The function prototype is:

```
float Inverse(float x) ;
```

Note: $x^{-1} = (x - 1)^0 - (x - 1)^1 + (x - 1)^2 - (x - 1)^3 + \ldots$

```
float Inverse(float x)
   {
   float coef[] = {1, -1, 1, -1, 1, -1, 1, -1} ;
         return Polynomial(x - 1, coef, 8) ;
   }
```

5. Write a function in C that calls the Polynomial function developed in problem 3 to compute an eight term Taylor series approximation to the trigonometric sine. Write a C program to test your function. The function prototype is:

```
float Sine(float radians) ;
```

Note: $\text{sine}(x) = x^1/1! - x^3/3! + x^5/5! - x^7/7! + \ldots\ldots$

```
float Sine(float radians)
   {
   float coef[] = {0, 1, 0, -1.0/3*2, 0, 1.0/5*4*3*2, 0, -1.0/7*6*5*4*3*2} ;
   return Polynomial(radians, coef, 8) ;
   }
```

6. Write a function in C that calls the Polynomial function developed in problem 3 to compute an eight term Taylor series approximation to the exponential. Write a C program to test your function. The function prototype is:

```
float Exponential(float x) ;
```

Note: $e^x = x^0/0! + x^1/1! + x^2/2! + x^3/3! + x^4/4! + \ldots$

```
float Exponential(float x)
   {
   float coef[] =
      {
      1.0, 1.0, 1.0/2, 1.0/3*2,
      1.0/4*3*2, 1.0/5*4*3*2, 1.0/6*5*4*3*2, 1.0/7*6*5*4*3*2
      } ;
   return Polynomial(x, coef, 8) ;
   }
```