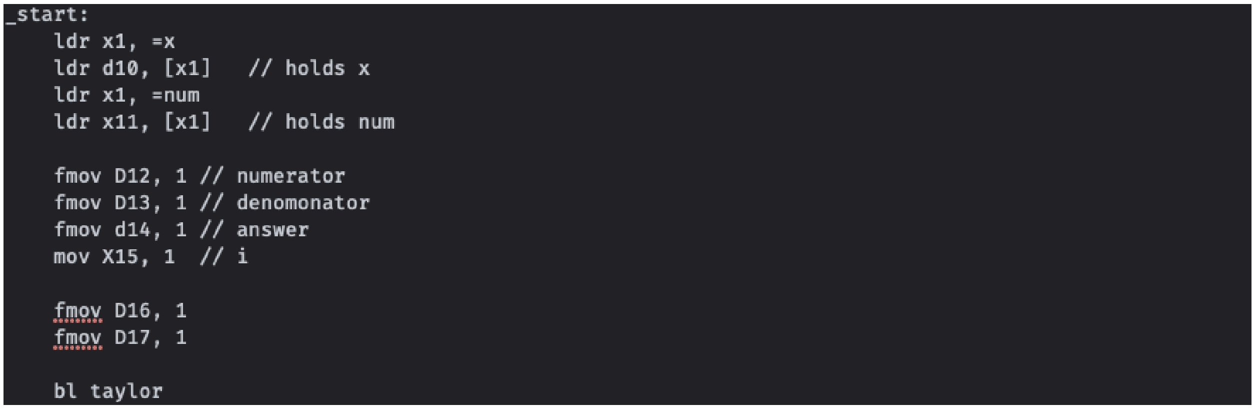
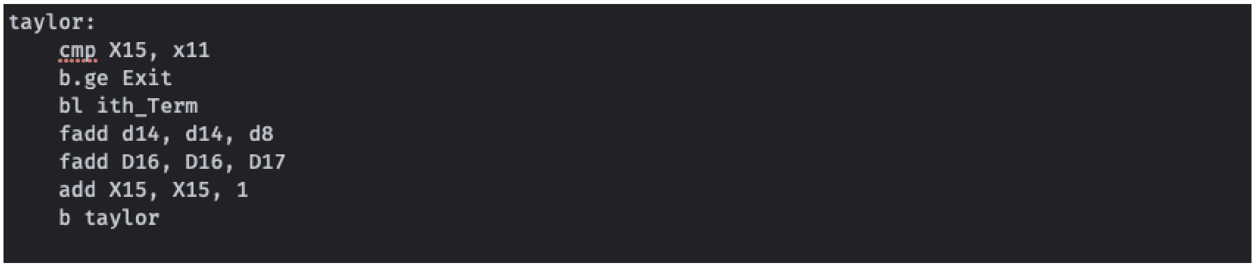


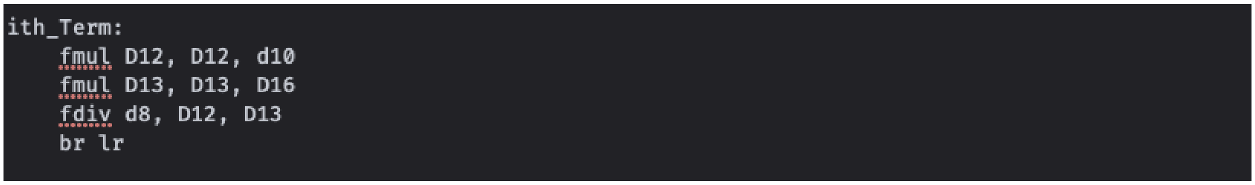
```
_start:
    ldr x1, =x
    ldr d10, [x1]   // holds x
    ldr x1, =num
    ldr x11, [x1]   // holds num

    fmov D12, 1 // numerator
    fmov D13, 1 // denomonator
    fmov d14, 1 // answer
    mov X15, 1  // i

    fmov D16, 1
    fmov D17, 1

    bl taylor
```

Here I load in the x and the num (the number of terms in the expression). I also set the numerator, denominator, answer, i, and temp registers. I then call teh taylor function



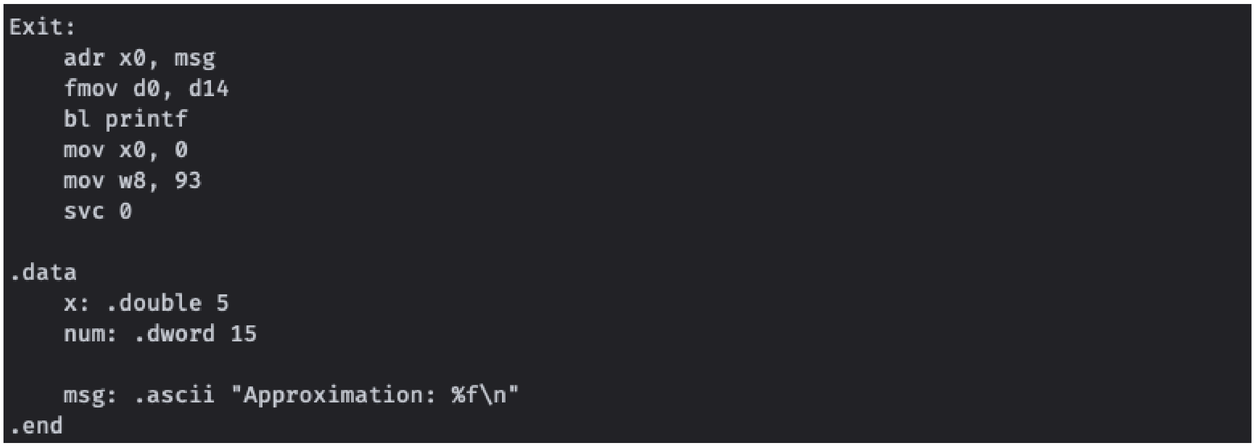
```
taylor:
    cmp X15, x11
    b.ge Exit
    bl ith_Term
    fadd d14, d14, d8
    fadd D16, D16, D17
    add X15, X15, 1
    b taylor
```

Here is where I loop through the iterations to compute and then add each term to the answer. Within each iteration i call “ith_Term” function to compute each ith term



```
ith_Term:
    fmul D12, D12, d10
    fmul D13, D13, D16
    fdiv d8, D12, D13
    br lr
```

Here is the “ith_Term” function that computes the math to calculate the ith term of the series.



```
Exit:
    adr x0, msg
    fmov d0, d14
    bl printf
    mov x0, 0
    mov w8, 93
    svc 0

.data
    x: .double 5
    num: .dword 15

    msg: .ascii "Approximation: %f\n"
.end
```

Here is when the exit call is called and I move the answer to D0 to print and then exit the program. This also shows my data section that holds x, num, and msg.