Breif outline of this is that im loading in the data (the array and teh length)

I then move to the outer loop where i check if teh sorting is done and load the next elem

```
Insertion_sort:
    LDR x10, =arr
    LDR x11, =length
    LDR x11,[x11]
    LSL x11, x11,#3
    ADD x11, x11, x10   // last elm addr
    MOV x12, x10
Outer_loop:
    CMP x12, x11
    B.GE printArray
    ADD x12, x12, #8    //next elm
    LDR x14, [x12]      // load next elm
    MOV x15, x12
```

In the inner loop i check if array[j] is greater then current and then also check to see if the loop fo the inner is done. If im still in the inner I make bring the elem to teh bigging

```
Inner_loop:
    SUB x15, x15, #8
    CMP x15, x10
    B.LT Outer_loop
    LDR x13, [x15]
    CMP x13, x14
    B.LE Outer_loop
    STR x13, [x15, #8]
    STR x14, [x15]
    B Inner_loop

_start:
    B Insertion_sort
```

```
.func printArray
printArray:
    LDR x10, =arr
    LDR x11, =length
    LDR x11, [x11]      //length in x11
    LSL x11, x11, #3
    ADD x11, x10, x11
    LDR x12, =Stack
    MOV SP, x12
    SUB SP, SP, #16
    STR x25, [SP]
    STR x26,[SP, #8]    //store x26 in stack offset 8
    MOV x25, x10        //move base array to x25
    MOV x26, x11        //move ending addr of array to x26
Loop:
    CMP x25, x26
    B.GE Done       //if curr item is last element, go to end
    LDR X0, =msg
    LDR X1, [x25]       //arr[i]
    BL printf
    ADD x25, x25, #8
    B Loop
Done:
    LDR X0, =endmsg
    BL printf           //print (newline character)
    LDR x25, [SP]
    LDR x26, [SP, #8]
    B End
.endfunc
```

Here im saving stuff onto teh stack to then print because print f kills all rregisters