

Jonathan Carbonneau

Selection Sort

I pledge my honor that I have abided by the Stevens Honor System.

```
//X19 array
//X20 array length
//x21 I iterator
//x22 tmp index pos1
//x23 counter for print
//x24 tmp store pos1
//x25 tmp store pos2
//x26 J iterator
```

Array and array size are defined in data.

Array: .dword 10,9,88,75,6,5,34,35,2,1 // Array A

Size: .dword 10 // Number f

The array is loaded into the x19 register.

The length is loaded into the x20 register.

The counter I is stored in x21 and set to zero.

The counter for the print statement is stored in x23 and set to zero.

```
ADR X20, Size // Load address of f
LDUR X20, [X20, #0] // Load the value of f
LSL X20, X20, #3
SUB X27, X20, #8
```

```
ADR X19, Array // Load base address of A
MOV X21, #0
MOV X23, #0
```

The following code is defined by the selection sort algorithm; the colors correspond to the parts in the pseudo code.

```

For int i=0; i<a.length-1; i++ {
    DO
        int pos=i;
        For int j = i + 1; j < a.length; j = j + 1 {
            DO
                If a[j]<a[pos] then pos=j;
            }
        }
        swap(a,pos1, pos2)
    }

swap(array a, pos1, pos2)
{
    temp = a[pos1];
    a[pos1] = a[pos2];
    a[pos2] = temp;
}

LoopI: // for(int i=0; i<a.length-1; i++)

    MOV X22, X21 int temp=i;
    ADD X26, X21, #8 //for(int j=i+1; j<a.length; j++)

LoopJ:

    LDR X24, [X19, X26]
    LDR X25, [X19, X22]
    CMP X24, X25 //if(a[j]<a[pos]) pos=j;

    BGT else
    MOV X22, X26
else:
    ADD X26, X26, #8
    CMP X26, X20
    BLT LoopJ
    BI Swap // swap()
    ADD X21, X21, #8
    CMP X21, X27
    BLT LoopI
    BI End

Swap:
    LDR X24, [X19, X21] //swap function
    LDR X25, [X19, X22]
    STR X24, [X19, X22]
    STR X25, [X19, X21]
    BR LR

```

The code then branches to the exit flag which contains a print statement that loops through the final array and prints the items. Then exits the code

End:

```
/* Print after value of A[0] */
```

Print:

```
ADR X0, msg
```

```
LDR X1, [X19, X23]
```

```
BL printf
```

```
ADD X23, X23, #8
```

```
CMP X23, X20
```

```
BLT Print
```

```
/* Exit the program */
```

```
MOV X0, #0
```

```
MOV X8, #93
```

```
SVC #0
```