

Learning Objective: To gain a fuller understanding of disk seek algorithms by implementing them.

Development: Implemented in GO to be run and tested on Eustis

Assignment: Implement the *first-come first-served scheduling, shortest seek time first, SCAN, C-SCAN, LOOK, C-LOOK* algorithms. Calculate and output the total amount of movements for the input cylinder requests and starting position. Review the examples below for details.

Input: Your program will read a file specified in the first command line parameter, `ARGV[1]`, which will be formatted as follows. Your program should ignore everything on a line after a `#` mark and ignore additional spaces in input.

```
Use sstf          # fcfs, sstf, scan, c-scan, look, or c-look
lowerCYL 00000    # valid lower cylinder number
upperCYL 00000    # valid upper cylinder number (> lower cylinder)
initCYL 00000     # initial cylinder position (0<initCYL<upperCYL)
cylreq 00000      # a single cylinder request
                  # where the lowerCYL < cylreq < upperCYL
cylreq 00000      # a single cylinder request
cylreq 00000      # a single cylinder request up to 20 requests
end
```

Note that there can be up to 20 cylinder requests. Each cylinder request must be bounded by the lower and upper cylinder values. In the event they are not, generate an error message and process the next cylinder request until reaching the end of the input file, signified by the **end** command.

Output: Generate the output formatted as shown below. (Send it to STDOUT.)

Given that the input file is *fcfs.in* (shown on the following page) the output file is shown below:

```
run diskScheduler.go fcfs.in
Seek algorithm: FCFS
  Lower cylinder:    0
  Upper cylinder:  4999
  Init cylinder:   1000
  Cylinder requests:
    Cylinder  2069
    Cylinder  1212
    Cylinder   544
    Cylinder  1618
    Cylinder  3681
  Servicing  2069
  Servicing  1212
  Servicing   544
  Servicing  1618
  Servicing  3681
  FCFS traversal count = 5731
```

This output shown above is derived from the following input file, *fcfs.in*.

```
Use fcfs
lowerCYL 0
UpperCYL 4999
initCYL 1000
cylreq 2069
cylreq 1212
cylreq 544
cylreq 1618
cylreq 3681
end
```

Submitting

Make sure your submission is named *diskScheduler.go* and submit this source file to Webcourses.