CP 386, Fall 2019
Assignment 3 (10% of the final grade)
(due Wednesday, December 4, at 11:30 pm)

Your task is to implement simplified `DiskScheduler` for an OS. It will schedule a sequence of requests to a hard drive, using the standard scheduling policies: `FCFS`, `SSTF`, `SCAN` and `C-SCAN` as described in your text. The sequence of requests to the hard drive will be provided via stdin with the following format:

```
k <cT> <direction> <tN1> <tN2>  ... <tNk>
```

where

- `k` is a positive integer representing the number of requests in the line
- `<cT>` is current track number (the track currently under disk reading head: an integer from the interval 0..1023))
- `<direction>` is "u" or "d" – current direction of disk head move (where "u" stands for "up" meaning that the head is moving in the direction of increasing track number, and "d" stands for "down").
- `<tNi>` requested track number (an integer from the interval 0..1023).
- For `C-SCAN` policy assume that reading direction is "up".

Implementation

You will be provided with

- `dscheduler.c`
- `dscheduler.h`
- `main.c`
- `makefile`

The provided code takes an argument which reads in the request from stdin and prints the results. **You only need to fill in the functions at the bottom of dscheduler.c which implement the actual policies**. The marker should be able to use an entirely different main and get the same results as you.

How to Test

- Compile using gcc -Werror -Wall -g -std=c99 -o DSCheduler dscheduler.c main.c
- Run
  ```
  ./DSCheduler FCFS < test_1.txt
  ```
  Replace FCFS with SSTF or SCAN or C-SCAN to test the other policies.

Marking

Your code will be tested against several test cases. Some of these will be provided on MLS prior to the due date and some will not. The expectation is that if your code passes the provided tests cases it should pass the additional ones, so if you only pass the provided test cases you will pass the assignment (although not with flying colors). Expect the additional tests cases to cover less obvious edge cases.

Note: Your code must compile using gcc -Werror -Wall -g -std=c99 -o DSCheduler dscheduler.c main.c

What to Submit

Upload **1 zip file** to the MLS dropbox named **Laurier ID #-CP386-A3.zip** . Example if your student ID # is 1723571113 you would submit **1723571113-CP386-A3.zip** Make sure your zip file **does not** contain folders. You may need to manually zip your files if using Eclipse to test them. See `https://mylearningspace.wlu.ca/d2l/lms/news/main.d2l?ou=313315` for examples.

Failure to name or zip your submission correctly will result in a mark of zero.

Sample Input & Output

```
5 100 u 12 54 150 17 41
```

Results for

- FCFS -> 12 54 150 17 41 Total head movement: 383
- SSTF -> 54 41 17 12 150 Total head movement: 226
- SCAN -> 150 54 41 17 12 Total head movement: 1934
- C-SCAN -> 150 12 17 41 54 Total head movement: 2000