1. Programming Task: Implement a Disk Scheduling Simulator

The objective of this project is to implement a simple disk scheduling simulator "**simdisk**" in
C/C++ on a UNIX-based platform.

SYNOPSIS: **simdisk** [**–h**] [**-n** *number-of-cylinders*] [**–d** *disk-shed-policy*] [**–i** *input_file*]

DESCRIPTION: **simdisk** is a simple disk scheduling simulator. It takes a sequence of disk cylinder I/O requests as an input, as well as the total number of cylinders on the disk. It performs the scheduling of incoming I/O requests using the disk scheduling policy specified by the user, and measures total number of cylinders travelled by the disk head.

1.1 OPTIONS:

–h **:** Print a usage summary with all options and exit.

–n *number-of-cylinders* **:** Set the total number cylinders on the disk. By default it should be 200. (Be careful that disk cylinder indexes would be between 0-199 in this case.)

– d *disk-shed-policy* **:** Set the disk scheduling policy. It can be either **FCFS** (First-come-first-served), **SSTF** (Shortest-seek-time-first), **C-SCAN**, or **LOOK**. The default will be **FCFS**. Assume at the t=t_0, the disk head was moving from 0 towards other end of the disk (i.e. cylinder **n**).

–i *input file* **:** Read the I/O request sequence from a specified file. If not
given, the sequence should be read from STDIN (ended with
ENTER).

1.2 OUTPUT:

simdisk will print out the sequence in which the I/O requests to cylinders will be served according to the scheduling algorithm selected. It will also print out the total number of cylinders travelled by the disk head when serving these requests.