Question 1:

In continuation with the last example in the README, use the shortest access-time or positioning time first (SATF) scheduler (`-p SATF`) instead of SSTF. It makes no difference in the total time taken for `7,30,8` workload.

a. Find a set of requests where SATF outperforms SSTF.
b. More generally, when is SATF better than SSTF?

Question 2:

For the set of requests: `10,11,12,13`, the default setup performs poorly. Try adding track skew to improve the performance (`-o skew`), where `skew` is an integer.

a. Given the default seek rate, what should the track skew be to maximize performance?
b. Try the same thing for different seek rates (e.g., `-S 2, -S 4`)? In general, write a mathematical formula to figure out the track skew for a given disk, if you are provided with the seek rate, rotational latency, and sectors per track?

Question 3:

Go through the source code of disk.py and implement either the C-SCAN **OR** the C-LOOK algorithm. Write down a set of requests that will work better with your algorithm compared with the default FIFO algorithm.
The set of requests should be able to run with `-p SCAN` option.

Question 4:

Read about the following Linux disk schedulers and write a brief description of each.

a. mq-deadline
b. bfq