1. Suppose that a disk drive has 1000 cylinders, numbered 0 to 999. The drive is currently serving a request at cylinder 150. The queue of pending requests, in FIFO order, is:

   32, 901, 447, 432, 637, 506, 851, 679, 173, 313

   Fill in the table below with the order the new requests are serviced.
   SSTF

Request	1	2	3	4	5	6	7	8	9	10
Cylinder	**173**	**313**	**432**	**447**	**506**	**637**	**679**	**851**	**901**	**32**

   Using the answer from above, how would you characterize the behavior of SSTF?
   Cylinders at the ends are visited less frequently than those in the middle

2. Definitions:
   a. Seek time is:
      The time necessary to move the disk arm to the desired cylinder
   b. Rotational Latency is:
      The time necessary for the desired sector to rotate to the disk head
   c. Transfer time is:
      The time necessary to read data from sector on platter
   d. **Transfer time** is usually not considered in disk scheduling because **the physical location of logical blocks is not disclosed.**

3. What are the tradeoffs with rereading code pages from the file system versus using swap space to store them?
   Time
   Performance
   Space
   Partition

4. Suppose that a disk drive has 5000 cylinders numbered 0 to 4999. The drive is currently serving a request at cylinder 2150 and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order is:

   2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

FCFS: **13011**
SSTF: **7586**
SCAN: **7492**
LOOK: **7424**
C-SCAN: **9917**
C-LOOK: **9137**

5. Consider a system where free disk space is kept in a free-space linked-list. Suppose the pointer to the free-space list is lost. Can the system reconstruct the free-space list?
   Yes, but it is very expensive via garbage-collection-like mechanisms

6. Consider a file system similar to the one used by UNIX (indexed allocation). How many disk operations are required to access the contents of a file at */dir1/dir2/file3*? Assume that none of the disk blocks is currently being cached.
   Total number of disk operations: **4**

7. You are asked to allocate a file according to either a File Allocation Table (FAT) or multi-level indexed allocation (UNIX inode - triply indirect).
   Assume that:
   - The file is 2000 disk blocks long
   - There are 4 KB per disk block
   - Each pointer in the FAT occupies 4 bytes
   - The first index block contains 15 entries (of which 12 are direct, and one each is singly indirect, doubly indirect, and triply indirect)
   - Every other index block contains 15 entries (may be indirect depending on the nesting level)
   - Each index block entry takes 4 bytes
   - Unused index blocks don't count in the total memory cost, though unused entries in partially filled index blocks do count.

   How many bytes (including allocation overhead) are used to lay out the file when using a:
   A. FAT file system
      8000
   B. UNIX-style file system
      8640

Now suppose that you wish to read the 1099'th block of the file. Assume each of the following counts as one search operation :

- Moving from one element to the next in a linked list
- Indexing into an index block
- Moving from index block to the next

How many searches are needed to read block 1099 when using the

C. Same FAT file system as above
   1099
D. Same UNIX-style file system as above
   4