Q1

1. What is the storage capacity of the disk (in bytes)? [1]

   4096*40*2*12000*2=15728640000bytes

2. How many disk heads does the disk head array contain? [1]

   one head per surface, and there are 8 surfaces. So, there are 8 disk head.

3. How many cylinders are there on the disk? [1]

   Cylinder is the set of tracks with same diameter on all surfaces. Number of cylinders is equal to number of tracks. There are 12000 cylinders.

4. What is the *maximum* time to read a single block? Show the values for each of the components of the read time. [3] (1 mark for each component)

   Max time to read a single block=Max seek time + Max rotational latency+ transfer time=12ms+11.11ms+0.56ms=23.67ms

   Max seek time=12ms

   Max rotational latency=60sec*1000ms/sec / 5400RPM≈11.11ms

   Transfer time=60sec*1000ms/sec /5400RPM/20blocks per tracks=0.56ms

5. What is the *average* time to read a single block? [3] (1 mark for each component)

   Average total time=average seek time + average rotational latency+ transfer time=4+5.56+0.56=10.12ms

   average seek time = 1/3 Max seek time=4ms

   average rotational time= 1/2 Max rotational time=5.56ms

   Transfer time=0.56ms

6. What is the *average* time to read an entire track, assume that the track must be read starting from a particular block? Show the values for each of the three components of the read time. [3] (1 mark for each component)

   Average total time=average seek time + average rotational latency+ transfer time=4+5.56+11.11=20.67ms

   average seek time = 1/3 Max seek time=4ms

average rotational time= 1/2 Max rotational time=5.56ms

Transfer time=60sec*1000ms/sec /5400RPM≈11.11ms

7. What is the *maximum* time to read two (unrelated) blocks? [2] (1 mark for answer, 1 mark for explanation)

   Total = 2*(Max seek time + Max rotational time + transfer time) =2*(23.67ms) =47.34ms

   In the worst case, disk head needs max seek time and disk needs max rotational time to find the first block. Assume the second block is far from the first block (in the origin position of disk head), then disk head need another max seek time and disk needs max rotational time to find the second block. So, the total time is two times of the *maximum* time to read a single block.

8. What is the *minimum* time to read two blocks? [2] (1 mark for answer, 1 mark for explanation)

   Min=2* transfer time=1.12ms

   Head disk point to the first block and two blocks are adjacent so that sum of seek time and rotational time is nearly zero. It only needs to transfer times.

9. What is the size in bytes of the largest data file that could be read without having to move the disk head? [1]

   Since the disk head cannot move, it can only read data on current track.

   There are 4 double sided patters and for each size there is a disk head. So, without moving disk head, 8 tracks of data could be read.

   There are 40 sectors per track and each sector has 4096bytes. So, the largest data file the disk head could read is 40*4096*8=1310720 bytes.

10. Why does defragmenting a hard drive improve performance? [1]

    Previous questions show that rotational time and seek time dominate the total time to read blocks. Defragmentation reduces the degree of fragmentation by moving small files to the close directory so that it takes less rotational time and seek time to read those files. Therefore, defragmenting a hard drive improve performance.

11. The question specifies that the disk head speed is assumed to be constant. Why is this assumption unrealistic? [1]

After read one file, the disk head needs to move to another place to read different files. The disk head moves with an initial speed zero so that it must accelerates before reaching the max speed. Constant speed is unrealistic for an moving object without initial speed.

Q2

Request	RAID 0	RAID 1+0	RAID 4	RAID 5
Read 0 to 31	1/4	1/8	1/4	1/4
Read 4,11,17 and 22	1/4	1/4	1/4	1/4
Read 1,9,14 and 17	3/4	1/2	3/4	1/2
Write 0 to 31	1/4	1/2	1/2	1/2
Write 4,11,17 and 22	1/4	1/2	5/4	5/4
Write 1,9,14 and 17	3/4	3/2	7/4	3/2
Cost	4	8	5	5

Q3

SCAN: head starts from one end of the disk and moves towards the other end. It will service requests in between one by one before reaching the other end. Then head will reverse the direction and the process continues as it continuously scans back and forth to access the disk.

LOOK: head starts from one end of the disk and moves towards the other end. It will service requests in between one by one and check if there are any requests pending ahead. If there is no request pending ahead, then head will reverse the direction and the process continues. Head will reverse once there is no requests pending in the direction of movement.

C-SCAN: like SCAN, head starts from one end of the disk and moves towards the other end. It will service requests in between one by one before reaching the other end. **But head will jump to the beginning position and repeats same actions.** C-SCAN is used to avoid starvation and services all the requests more uniformly

C-LOOK: like LOOK, head starts from one end of the disk and moves towards the other end. It will service requests in between one by one. **But head will jump to the beginning position and repeat moves towards to the other end once there are no requests pending in the direction of movement.** C-LOOK is used to avoid starvation and services all the requests more uniformly