Q1)

Total resources

5 tape drives, 3 modems, 5 printers

currently allocated

	Tapes	modems	printers
P_0	0	0	1
P_1	0	0	1
P_2	1	0	1
P_3	2	2	2
P_4	1	0	0

STILL required

	T	M	P
P_0	5	2	1
P_1	0	1	0
P_2	1	2	3
P_3	1	1	1
P_4	4	2	4

Available

T	M	P
1	1	0

Q1) cont.

Available: T M P
1 1 0

Run P_1 (requires 1 modem)

Available: T M P
1 0 0

P_1 completes and releases 1 modem and 1 printer

Available: T M P
1 1 1

Run P_3, completes and releases 3 of each

Available: T M P
3 3 3

Run P_2, completes and releases 2T, 2M, 4P

Available: T M P
4 3 4

Run P_4: releases 5T, 2M, 4P

Available: T M P
5 3 4

Run P_0, releases 5T, 2M, 2P

Available:	T	M	P
	5	3	5

No dead-locks, state is safe

(Q2)

Reference order: A, B, A, B, A, A, C, D, D, B, D, E, A, D, A

FIFO with 3 frames = 6 replacements

F1	F2	F3
A		
A	B	
A	B	C
D	B	C
D	E	C
D	E	A

FIFO with 4 frames

F1	F2	F3	F4
A			
A	B		
A	B	C	
A	B	C	D
E	B	C	D
E	A	C	D

6 replacements

FIFO with 5 frames

F1	F2	F3	F4	F5
A				
A	B			
A	B	C		
A	B	C	D	
A	B	C	D	E

5 replacements

Q2)

Ref: A, B, A, B, A, A, C, D, D, B, D, E, A, D, A

Second Chance with 3 frames

F_1	F_2	F_3
A_1		
A_1	B_1	
A_1	B_1	C_1
D_1	B_0	C_0
D_1	B_1	C_0
D_1	B_0	E_1
D_1	A_1	E_1
D_1	A_1	E_1

6 replacements

Q2)

Second chance with 4 frames

F_1	F_2	F_3	F_4
A_1			
A_1	B_1		
A_1	B_1	C_1	
A_1	B_1	C_1	D_1
E_1	B_0	C_0	D_0
E_1	A_1	C_0	D_0
E_1	A_1	C_0	D_1

6 replacements

Q2)

Second chance with 5 frames

F_1	F_2	F_3	F_4	F_5
A_1				
A_1	B_1			
A_1	B_1	C_1		
A_1	B_1	C_1	D_1	
A_1	B_1	C_1	D_1	E_1

5 replacements

Q2)

clock with 3 frames

F_1	F_2	F_3	
A_1			(1)
A_1	B_1		(2)
A_1	B_1	C_1	(3)
D_1	B_0	C_0	(1)
D_1	B_1	C_0	
D_1	B_0	E_1	(3)
D_0	A_1	E_1	(2)
D_1	A_1	E_1	

6 replacements

Q2)

clock with 4 frames

F_1	F_2	F_3	F_4	
A_1				(1)
A_1	B_1			(2)
A_1	B_1	C_1		(3)
A_1	B_1	C_1	D_1	(4)
E_1	B_0	C_0	D_0	(1)
E_1	A_1	C_0	D_0	(2)
E_1	A_1	C_0	D_1	

6 replacements

(Q2)

clock with 5 frames

F_1	F_2	F_3	F_4	F_5	
A_1					(1)
A_1	B_1				(2)
A_1	B_1	C_1			(3)
A_1	B_1	C_1	D_1		(4)
A_1	B_1	C_1	D_1	E_1	(5)

5 replacements

Q3)
Pointers per block = 8K/4 = 2048 blocks
Total storage = (15 * 8KB) + 5(2048 * 8KB) + 4(2048 * 2048 * 8KB)

1) *What is the size (in bytes) of the smallest file the requires use of an indirect block pointer?*
   The smallest file too big for 15 direct block pointers
   Direct block pointer = 8KB = 8192B
   So the answer is: 8192B * 15 + 1 = 122881B

2) *What is the size (in bytes) of the smallest file the requires use of a double-indirect block pointer?*
   The smallest file too big for 5 indirect block pointers and 15 direct block pointers
   Indirect block pointer = 2048*8KB = 16,384KB = 16,777,216B
   Answer: 8192B*15 + 16,777,216B*5 + 1 = 84,008,961B

3) *What is the size (in bytes) of the largest file supported by this system?*
   Total storage
   (15*8KB)+5(2048*8KB)+4(2048*2048*8KB) = 134,299,769KB = 137,522,962,432B

4) *What is the size (in bytes) of the largest file that would be supported by this system if it had 64-bit block numbers instead of 32-bit block numbers?*
   Pointers per block would change to 8K/8 2048^2 = 4,194,304KB
   (15*8KB) + 5(2048^2 * 8KB) + 4(2048^2 * 2048^2 * 8KB) = 5.6295x10^14KB = 5.7646^17B

Q4)

1) Cluster takes time T to complete the program
   One computer takes time T*0.68*10 + T*0.32 = 7.12T
   Therefore, one computer takes 7.12 times longer than the cluster.

2) Cluster takes time T to complete the program
   One computer takes time T*0.68*10+T*0.3*4+T*0.02 = 8.02T
   Therefore, one computer takes 8.02 times longer than the cluster

3) Cluster takes time T to complete the program
   One computer takes time T*0.16*10+T*0.5*4+T*0.34 = 3.94T
   Therefore, one computer takes 3.94 times longer than the cluster

Q5)
Computers are one of humanity's greatest achievements, they have driven our capabilities beyond what our natural selves can accomplish. Amidst the COVID pandemic, computers have become an even greater workhorse behind modern society. However, with great power comes great responsibility. As computers evolved and became more accessible to the world, they came in contact with malicious individuals and groups who seek to abuse these tremendous technologies for their own agenda.

Today we know computer viruses as bits of code that traverse the internet seeking machines to spy on or perform other ill-intended activity. But how did we get here?

One of the earliest known viruses was the Creeper virus which infected Digital Equipment Corporation's computers in 1971. This virus was eventually combated and deleted by the Reaper anti-virus virus which was developed by the same author Ray Tomlinson to spread and stop the creeper virus. These programs were mostly experimental and did not damage the systems. It wasn't until the 80s when viruses started to become serious threats.

As viruses were being developed and distributed throughout the 80s, the need for anti-viruses became more and more of a necessity. Bernd Fix in 1987 developed what was arguably the first anti-virus produced. Then also in 1987 Andreas Luning and Kai Figge launched anti-virus software for Atari ST platform. In 1987 John McAfee founded McAfee (a leading system security company still to this day) and released VirusScan which is one of today's most popular anti-virus software systems.

Virus and anti-virus technologies are a continuous game and cat and mouse. A new virus will typically be developed specifically to avoid major anti-virus software detection systems. Then once it's released and made known to anti-virus companies/groups, it will be analysed and the next anti-virus version will be capable of detected that new type of virus. Then repeat. However, sometimes it's not as straight forward as this.

Some viruses will adopt a cryptographic or polymorphic mechanism to hide themselves in a manner that's extremely difficult to detect. Encryption makes the virus code appear meaningless and non-sensical. However, when the time comes to use the virus, a decryption function runs through the code and produces a virus which can then be executed without detection. Polymorphic code may be used to continually mutate the virus every time it runs, changing the appearance of the code but maintaining the logical algorithms as they were when originally developed.

These kinds of viruses are extremely difficult to detect. However, modern anti-virus software companies have adapted to these challenges and developed pattern analysis algorithms to search for the underlying patterns which these virus maintain between mutations. However, viruses can then use metamorphic code to produce a logical equivalent version of itself under some other presentation.

As you can see, the history of viruses and anti-virus software is truely just a game of cat and mouse with extra steps. The technologies become more and move advanced as time goes on, and sometimes anti-virus software doesn’t always protect its host machine. So it’s a good idea always maintain a cautious attitude towards any unfamiliar websites, downloads, or other online resources.