2018 (Fall).

Section "A"

Q.No.1

Ans. A thread is a sequence of instructions that can be executed independently within a program. A program can have multiple threads running concurrently, each executing a different set of instructions.

Multithreading is the ability of a program or an operating system to enable more than one user at a time without requiring multiple copies of the program running on the computer.

Q.No.2.

Ans. An Operating System is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output and controlling peripheral devices.

Operating System as a resource manager provides for an orderly and controlled allocation of the processors, memories and I/O devices among the various programs in the bottom-up view.

Q.No.3

Ans The difference between Interrupt driven I/O and DMA are:-

	Interrupt Driven I/O	DMA
i)	It requires the CPU to be involved in the data transfer.	It doesn't require the CPU to be involved in the data transfer.
ii)	Interrupt transfers can be slow and less effiuent as compared to DMA.	DMA tranefers can be faster and more efficient than interrupt driven I/O

Q No. 4

Ans. Contiguous memory allocation is basically a method in which a single contiguous section/part of memory is allocated to a process or file needing it. It is faster in execution as compared to Non-Contiguous memory allocation.

Non-Contiguous memory allocation is basically a method on contrary to contiguous allocation method, allocates the memory space present in different locations to the process as per its requirements.

Q No. 5.

Ans The CPU time is the time taken by the CPU to execute the process.

The total amount of time required by the CPU to execute the whole process is called Burst Time.

Q. No. 6.

Ans Files are used for all input and output (I/O) of information in the operating system, to standardize access to both software and hardware.

A directory is a unique type of file that contains only the information needed to access files or other directories.

Q. No. 7.

Ans The main difference between multiprogramming with a fixed number of task (MFT) and multiprogramming with variable number of task (MVT) is the level of flexibility they provide in allocating resources to processes. MVT allows a system to dynamically adjust the number of processes running, while MFT provides a more static environment with a predetermined number of processes.

Q.No.8

Ans Distributed system is a collection of autonomous computer systems that are physically separated but are connected by a centralized computer network that is equipped with distributed system software.

The advantages over centralized system are:-

i) Scalability
ii) Availability
iii) Fault tolerance
iv) Performance
v) Security

Q.No.9

Ans. The necessary conditions of Deadlock are:-

i) Mutual Exclusion
ii) Hold and wait
iii) Non-Preemptive
iv) Circular wait.

Q.No.10

Ans. The goals of security system in a operating system are to protect the system and its resources from unauthorized access, prevent data loss or theft, and ensure the confidentiality, integrity, and availability of data. Some goals of a security system are:-

i) Authentication
ii) Authorization
iii) Confidentiality
iv) Integrity
v) Availability
vi) Auditability.

Some threats to security systems include:-

(i) Malware

(ii) Hacking

(iii) Social engineering

(iv) Insider threats

(v) Denial of service attacks.

Q.No.12

Ans. Semaphore is a synchronization technique used to control access to a shared resource in a multi-threaded or multi-process environment. The critical section problem arises when multiple threads or processes attempt to access the same shared resource simultaneously, which may result in data inconsistency or race conditions.

To solve the critical section problem using semaphores, the following steps are taken:

i) Declare a semaphore variable and initialize it to 1.

ii) In each thread or process, use the semaphore wait() function to acquire the semaphore before accessing the shared resource.

iii) After accessing the shared resource, use the semaphore signal() function to release the semaphore, allowing other threads or processes to access the resource.

By using semaphores to control access to shared resources, we can ensure that only one thread or process accesses the critical section at a time, thereby preventing data inconsistency or race conditions.

Semaphore is a powerful tool and can be used to solve a wide range of synchronization problems. However, it is not always the best solution for the critical section problem. Other synchronization mechanisms, such as locks or monitors, may be more appropriate depending on the specific requirements of the system.

To solve this problem using semaphores, we can define two semaphores, empty and full, that represent the number of empty and full slots in the buffer, respectively. We also need a mutex semaphore to ensure that only one process can access the buffer at a time. Here's how the algorithm would work:

Date
Page

Producer:

i. Wait on the empty semaphore to ensure that there is at least one empty slot in the buffer.
ii. Wait on the mutex semaphore to gain exclusive access to the buffer.
iii. Insert a data item into the buffer.
iv. Signal the mutex semaphore to release the buffer.
v. Signal the full semaphore to indicate that there is one more full slot in the buffer.

Consumer:-

i. Wait on the full semaphore to ensure that there is at least one full slot in the buffer.
ii. Wait on the mutex semaphore to gain exclusive access to the buffer.
iii. Remove a data item from the buffer and process it.
iv. Signal the mutex semaphore to release the buffer.
v. Signal the empty semaphore to indicate that there is one more empty slot in the buffer.

In this algorithm, the empty semaphore ensures that the producer doesn't insert data into a full buffer, while the full semaphore ensures that the consumer doesn't remove data from an empty buffer. The mutex semaphore ensures that only one process can access the buffer at a time, preventing race conditions and other synchronization issues.

Q.No.13

Ans. The implementation of an operating system can give rise to several security issues. Some of these issues include :-

i) Vulnerabilities in the OS code:-

Like any software, an OS can have vulnerabilities that can be exploited by attackers. These vulnerabilities can lead to attacks such as buffer overflows, privilege escalation, denial-of-service and others.

ii) Malicious software:-

An OS can be infected by malware such as viruses, worms, and Trojans, which can compromise the security of the system. Malware can steal sensitive information, modify or delete data, or damage the system.

iii) Insecure configurations:-

An improperly configured OS can make the system more vulnerable to attacks. Examples of insecure configurations include weak passwords, unnecessary services enabled, and unpatched software.

iv) User errors:-

Human errors such as using weak passwords, sharing login credentials, and clicking on suspicious links can lead to security breaches in an OS.

v) Lack of access control:-

A lack of access control mechanisms in an OS can allow unauthorized access to sensitive data and resources. This can be exploited by attackers to steal data, modify files or cause system damage.

Cryptography is the practice of securing information by converting it into a form that cannot be read by unauthorized individuals. The process of cryptography involves several steps:-

i) **Plaintext:**

The original message that needs to be encrypted is called plaintext.

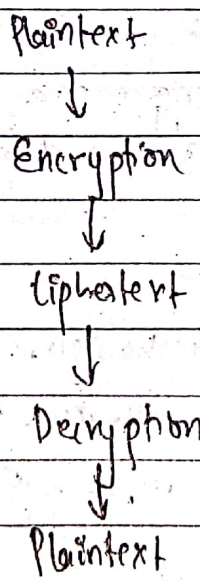
ii) **Encryption:**

The process of converting the plaintext into an unreadable form is called encryption. This is done using an encryption algorithm and a secret key. The encrypted message is called ciphertext.

iii) **Decryption:**

The process of converting the ciphertext back into plaintext is called decryption. This is done using a decryption algorithm and the secret key.

Here's a diagram to illustrate the cryptography process:



In this diagram, the plaintext is first encrypted using an encryption and a secret key to produce ciphertext. The ciphertext can then be transmitted over an insecure channel without being read by unauthorized individuals. The ciphertext is then decrypted using a decryption algorithm and the secret key to produce the original plaintext. This ensures the confidentiality and integrity of the original message.

Q.No.16.

Ans. Deadlock is a situation that can occur in a computer system when two or more processes are waiting for each other to release resources. To eliminate deadlock, we need to satisfy the necessary conditions.

The necessary condition for the elimination of deadlock is to ensure that at least one of the necessary conditions for deadlock is not met. These conditions are:-

i) Mutual Exclusion:

Resources must be shareable among processes.

ii) Hold and Wait:

A process must acquire all the resources it needs at once, rather than acquiring them one at a time while holding resources.

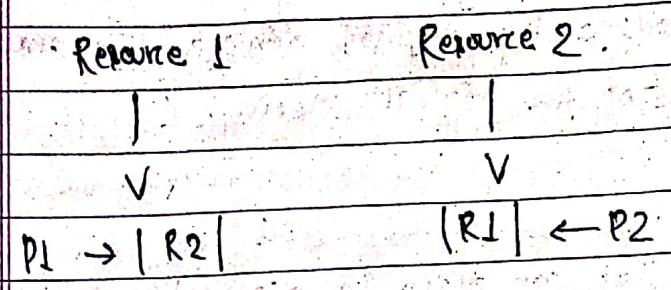
iii) No Preemption:

A process must release all its resources if it is unable to acquire a new one.

iv) Circular Wait:-

Resources must be ordered, and processes must request them in a specific order.

The figure below shows an example of how the necessary condition can be met by ordering resources and requiring processes to request them in a specific order.



In this example, P1 requests Resource 2 first, and then Resource 1, while P2 requests Resource 1 first and then Resource 2. This ordering ensures that there can be no circular wait, and deadlock is prevented.

Q.No.17.

Ans. Public and private key cryptography is an important technique used to secure communications over the internet and other communication channels. In this technique, a pair of keys is used - one for encryption and one for decryption. The encryption key is public and can be shared with anyone, while the decryption key is private and must be kept secret.

Cryptography is the practice of securing communications from unauthorized access or interception. Cryptography is used to protect data from eavesdropping, tampering, and forgery. Cryptography involves using mathematical algorithms and keys to encrypt and decrypt messages. The use of cryptography is crucial in secure communication, e-commerce, and online transactions.

The primary advantage of public and private key cryptography is that it provides a secure way for two parties to communicate over an insecure channel without sharing their private keys. The sender can use the receiver's public key. Similarly, the receiver can use the sender's public key to encrypt a response, which can only be decrypted using the sender's private key. This process provides authentication, confidentiality, and non-repudiation.

Public and private key cryptography has many applications, including secure communication over the Internet, digital signatures, and secure access control. Some examples of where public and private key cryptography is used include:

i) Secure communication:

Public and private key cryptography is used to secure communication over the internet, such as secure email and online transactions.

ii) Digital signatures:

Public and private key cryptography is used to provide digital signatures, which provide a way to verify the authenticity of a document

or message.

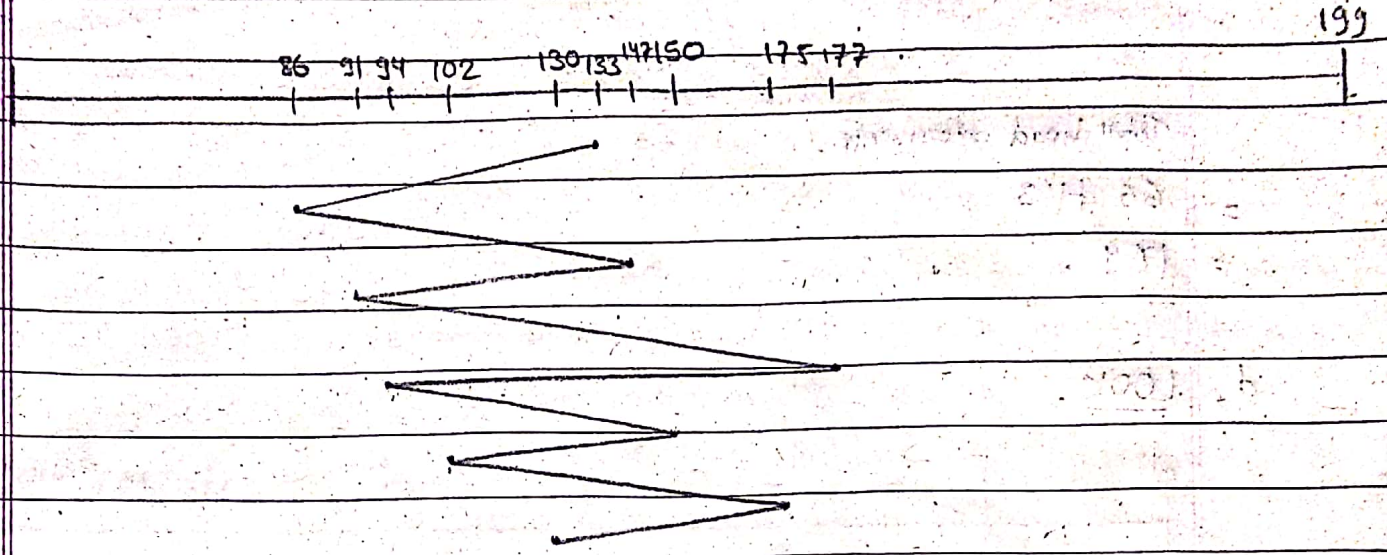
iii) Secure access control: Public and private key cryptography is used to provide secure access control, such as in secure authentication systems.

In summary, public and private key cryptography is an important technique used to secure communications over the internet and other communication channels. Cryptography is used to protect data from eavesdropping, tampering and forgery. The use of public and private key cryptography provides authentication, confidentiality, and non-repudiation, making it a crucial aspect of secure communication and e-commerce.

Q.No.11. Given,

Ans. 86, 147, 91, 177, 94, 150, 102, 175, 130.

a) FCFS.

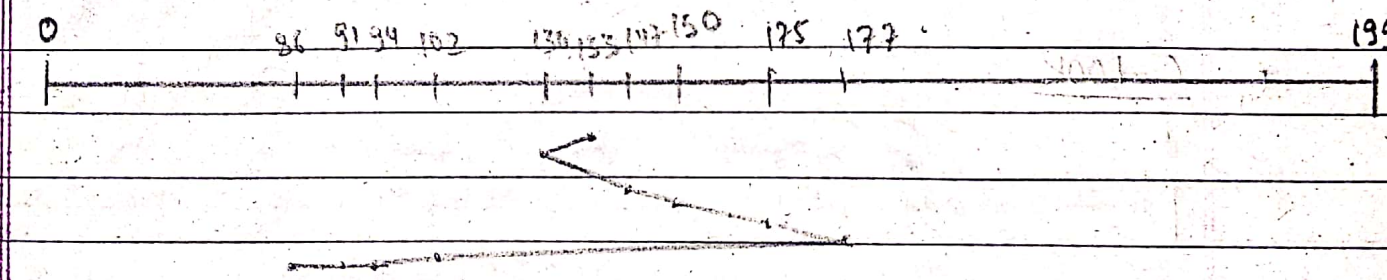


Total head movement.

= 47 + 61 + 56 + 86 + 83 + 56 + 48 + 73 + 45

= 555

b, SSTF.

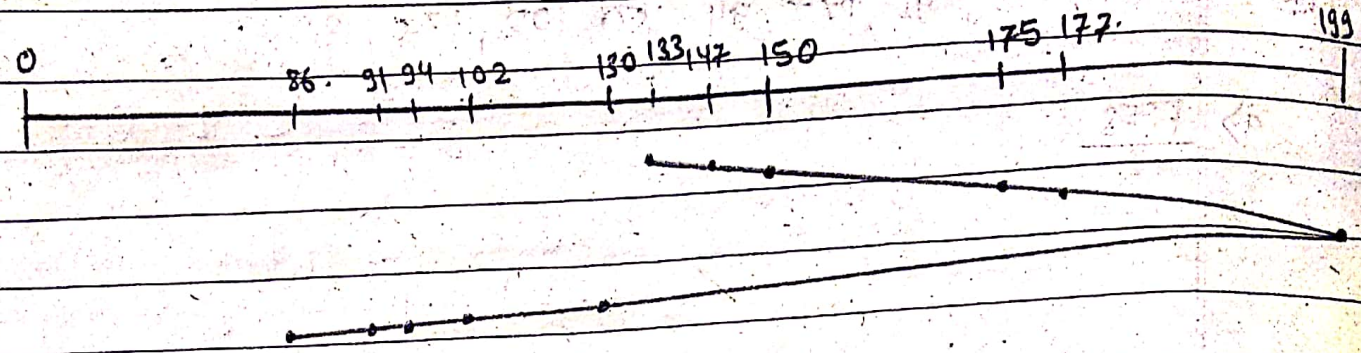


Total head movement.

= 3 + 47 + 91

= 141.

c, SCAN :

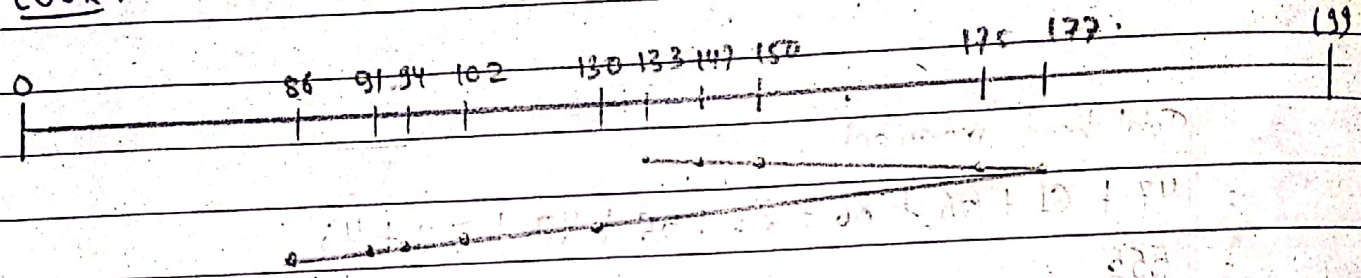


Total head movements.

= 66 + 113

= 179.

d, LOOK.

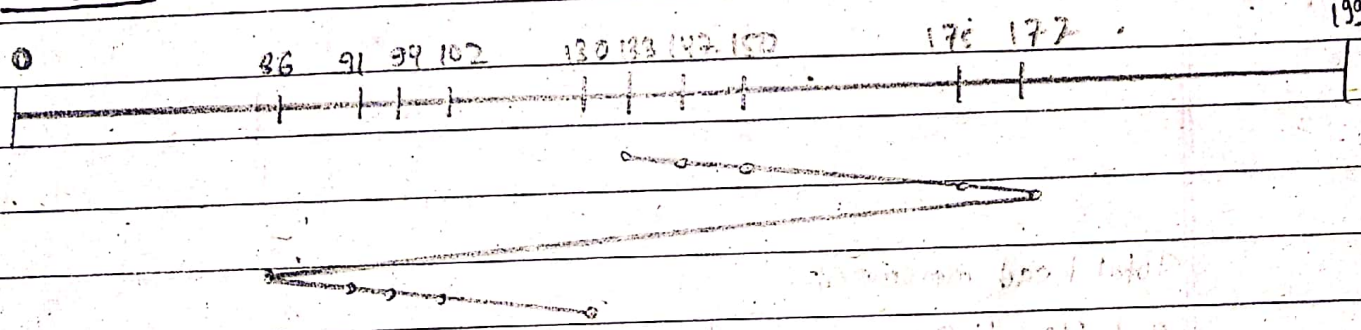


Total head movements.

= 44 + 91

= 135.

e, C-LOOK.



Total head movements.

= 44 + 91 + 44

= 179.

Q No.14. Given,

Ans.

Process	Arrival Time	CPU time	Priority
A	0	4	2(L)
B	1	1	4
C	2	3	6
D	3	5	10
E	4	1	8
F	5	4	13(H)
G	6	6	9

(TAT) Turnaround Time = Completion Time − Arrival Time

(WT) Waiting Time = Turnaround Time − Burst Time

a) FCFS :-

A	B	C	D	E	F	G
0	4	5	8	13	14	18	24.

Process.	TAT	WT
A	4	4-4=0
B	4	4-1=3
C	6	6-3=3
D	10	10-5=5
E	10	10-1=9
F	13	13-4=9
G.	18	18-6=12
	ATAT=65/7	AWT=41/7
	=9.28	=5.85.

b, SJF (preemptive)

A	B	A	E	C	F	D	G
0	1	2	5	6	9	13	18	24

Process	TAT	WT
A	5 - 0 = 5	5 - 4 = 1
B	2 - 1 = 1	1 - 1 = 0
C	9 - 2 = 7	7 - 3 = 4
D	18 - 3 = 15	15 - 5 = 10
E	6 - 4 = 2	2 - 1 = 1
F	13 - 5 = 8	8 - 4 = 4
G	24 - 6 = 18	18 - 6 = 12
	ATAT = 56/7 = 8	AWT = 32/7 = 4.57

c, Round-robin (quantum = 3)

A	B	C	D	A	E	F	G	D	F	G
0	3	4	7	10	11	12	15	18	20	21	24

Process	TAT	WT
A	11 - 0 = 11	11 - 4 = 7
B	4 - 1 = 3	3 - 1 = 2
C	7 - 2 = 5	5 - 3 = 2
D	20 - 3 = 17	17 - 5 = 12
E	12 - 4 = 8	8 - 1 = 7
F	21 - 5 = 16	16 - 4 = 12
G	24 - 6 = 18	18 - 6 = 12
	ATAT = 78/7 = 11.14	AWT = 54/7 = 7.714

e, SJF (non-preemptive)..

A	B	E	C	F	D	G
0	4	5	6	9	13	18	24.

Process	TAT	WT
A	4-0 = 4	4-4 = 0
B	5-1 = 4	4-1 = 3
C	6-2 = 4	4-3 = 1
D	9-3 = 6	6-5 = 1
E	13-4 = 9	9-1 = 8
F	18-5 = 13	13-4 = 9
G	24-6 = 18	18-6 = 12
	ATAT = 58/7	AWT = 34/7
	= 8.28	= 4.85

Q.No.15. Given,

Ans. page frame = 4.

reference string = 3, 3, 5, 4, 7, 1, 5, 5, 1, 4, 3, 7, 6, 3, 4, 2

a) LRU Page replacement:

3	3	5	4	7	1	5	5	1	4	3	7	6	3	4	2
3		3	3	3	1					1	1	6			6
		5	5	5	5					5	7	7			2
			4	4	4					4	4	4			4
				7	7					3	3	3			3

page fault : 9

page hit : 7.

b) Optimal page replacement

3	3	5	4	7	1	5	5	1	4	3	7	6	3	4	2
3		3	3	3	3						3	3			3
		5	5	5	5						7	7			2
			4	4	4						4	4			4
				7	1						1	6			6

page fault = 8

page hit : 8.

c, FIFO.

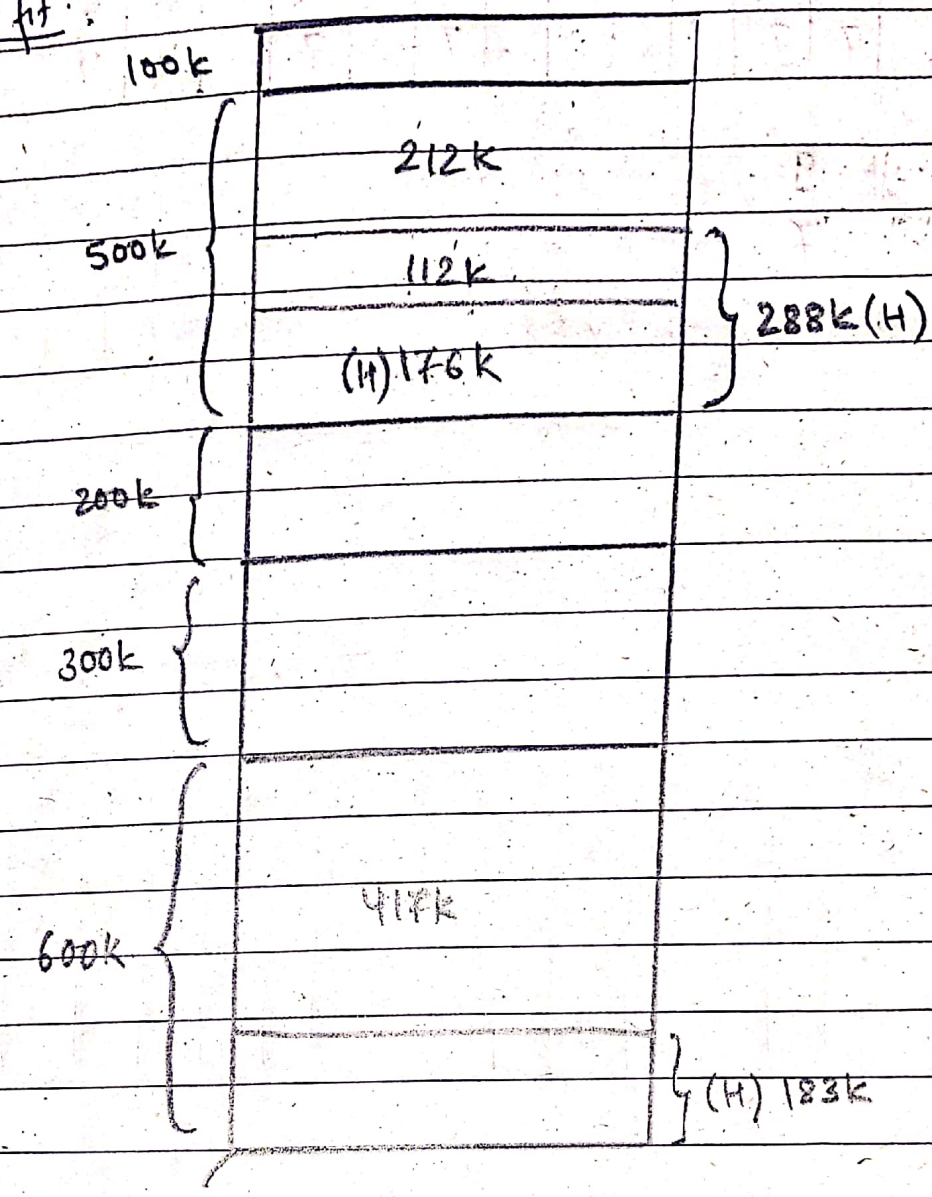
3	3	5	4	7	1	5	5	1	4	3	7	6	3	4	2
3		3	3	3	1					1		1		1	2
		5	5	5	5					3		3		3	3
			4	4	4					4		6		6	6
				7	7					7		7		4	4

page fault : 9
page hit : 7.

Q. No. 18

a) Ans. Given,
Memory partition ⇒ 100k, 500k, 200k, 300k, 600k.
Process ⇒ 212k, 417k, 112k, 426k

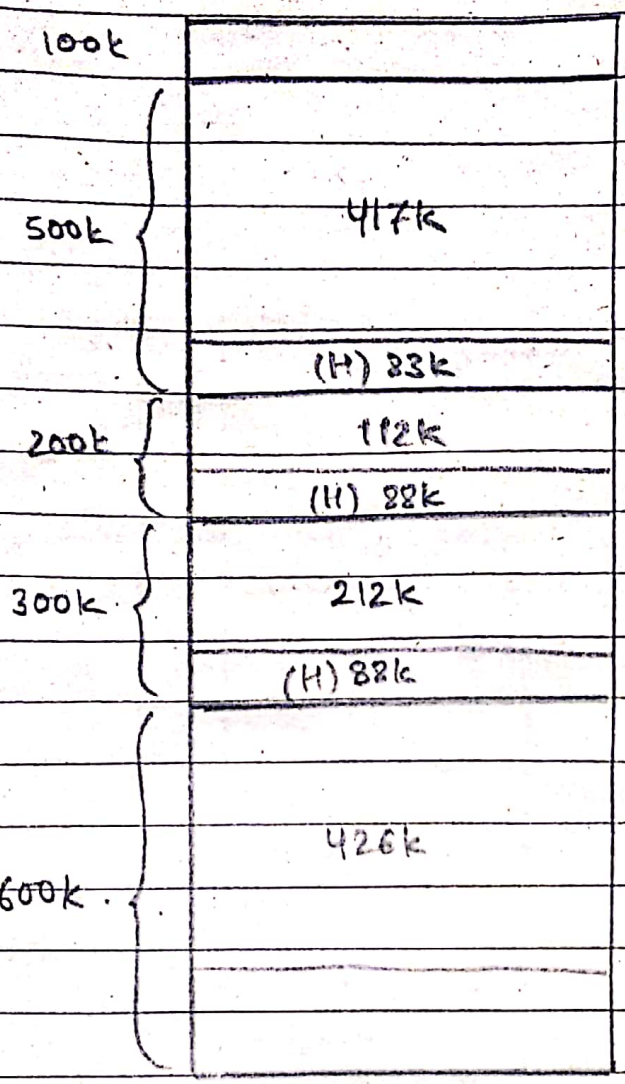
i) First fit:



* 426k cannot get memory.

Memory wastage = 1700k − (212 + 112 + 417)
= 1700 − 741
= 959 k.

d) Best Fit :-

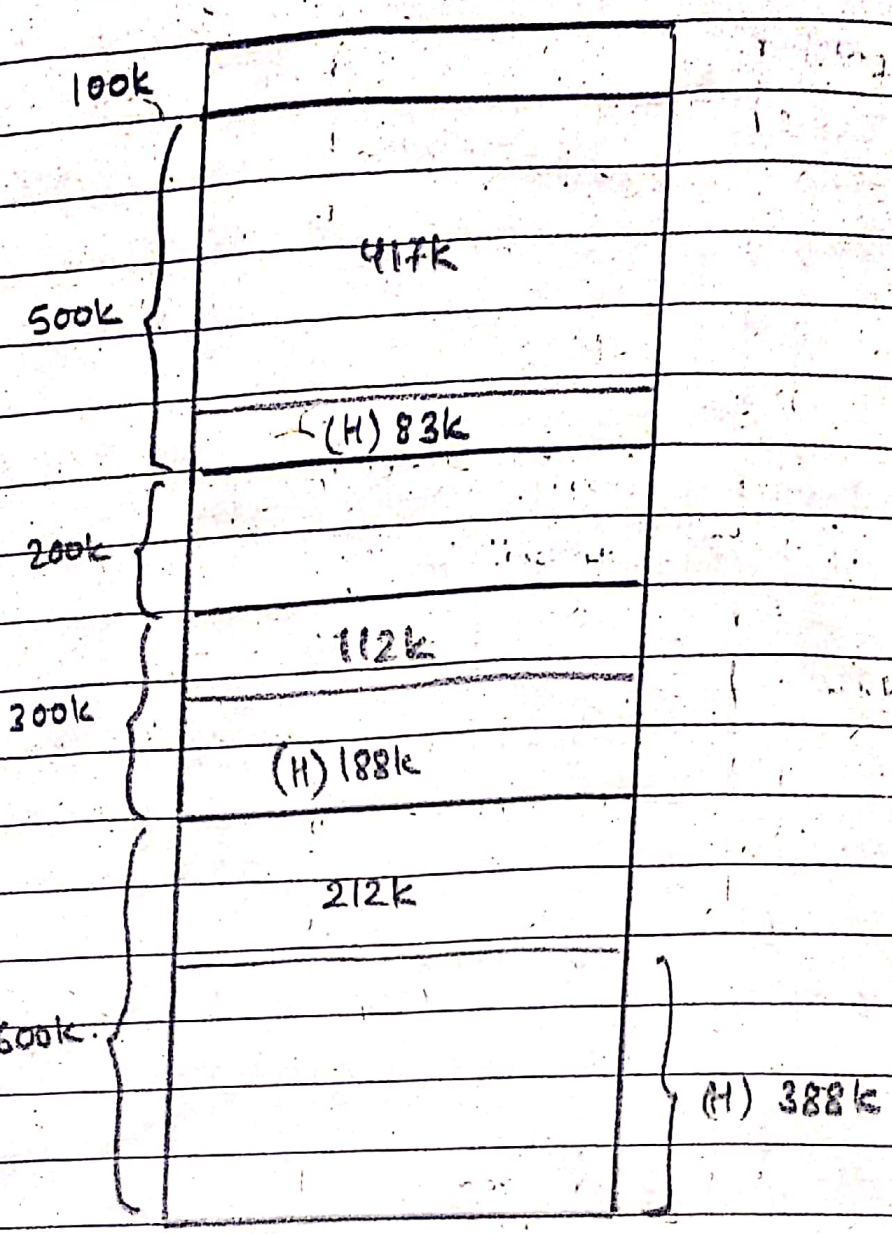


∴ Memory Wastage = 1700 − (417 + 112 + 212 + 426)

= 1700 − 1167

= 533k.

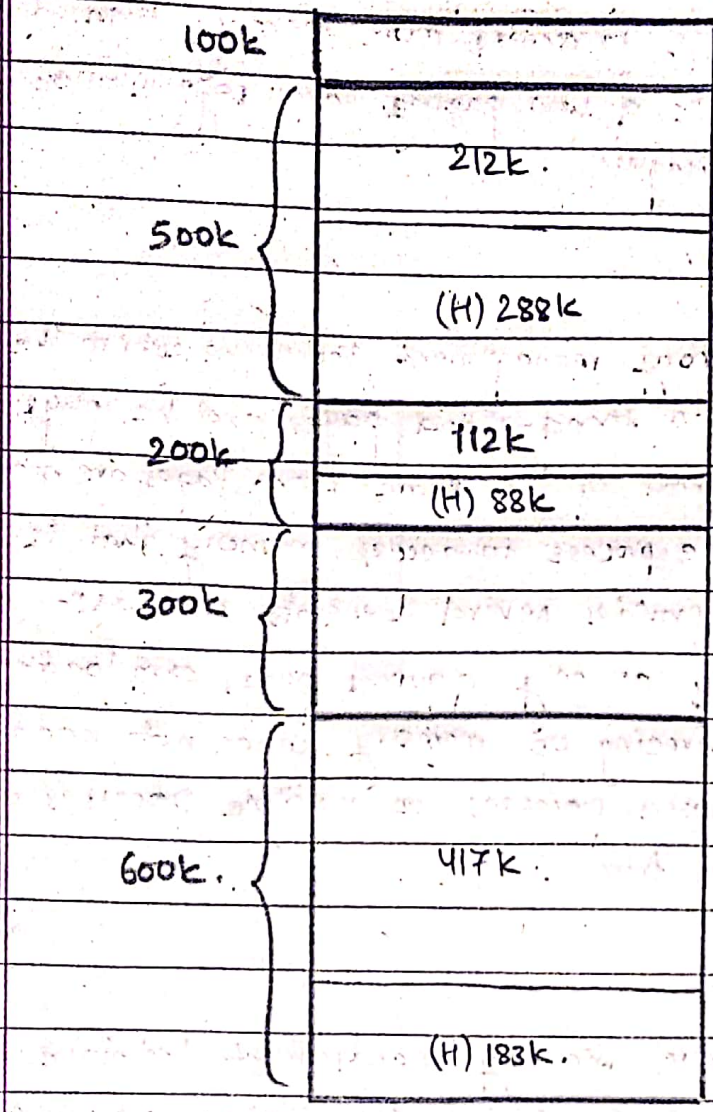
iii) Worst Fit:



* 426k cannot get memory.

Memory wastage = 1700 − (417 + 112 + 212)
= 1700 − 741
= 959k

iv, Next Fit:



↑ 426k cannot get memory

Memory wastage = 1700 − (212 + 112 + 417)
= 1700 − 741
= 959 k.

b>

Ans. Paging and Segmentation are both techniques used in virtual memory management to allow programs to access more memory than is physically available. Here is an analysis of how paging and segmentation work & their advantages and disadvantages:-

- **Paging:**

Paging is a virtual memory management technique where the process's address space is divided into fixed-sized pages, and the physical memory is divided into frames of the same size. Pages are mapped onto frames, allowing the process to access memory that is not physically present. Paging provides several benefits such as:-

- Efficient use of memory as only required pages are loaded into memory when needed, freeing up memory when not needed.
- Sharing of memory between processes, as multiple processes can share the same physical frame.

- **Segmentation:**

Segmentation is a virtual memory management technique where the process's address space is divided into logical segments, such as code, data, and stack. Each segment is of variable size and may have different access permissions. These segments are then mapped onto the available frames in physical memory. Segmentation provides several benefits such as:-

- Efficient use of memory by allowing the allocation of memory to match the specific needs of a process.
- Protection of memory, as each segment can have different access permissions, preventing unauthorized access to sensitive data.

In summary, both paging and segmentation are effective virtual memory management techniques that allow processes to access memory beyond what is physically available. Paging is more suited to situations where memory usage is unpredictable, while segmentation is more suited to situations where memory needs can be anticipated and allocated accordingly.

c)

Ans. Fragmentation in memory management occurs when there is unused memory due to the way memory is allocated and deallocated. This can lead to inefficiencies in memory usage and can limit the number of processes that can be accomodated in the memory. There are two types of fragmentation : internal and external.

Different memory allocation techniques in operating system can be affected by fragmentation in different ways.

- Fixed partitioning can suffer from wasted space in partitions that are larger than necessary for a process, limiting the no. of processes that can be accomodated.
- Variable partitioning can suffer from internal fragmentation when a process is allocated more memory than it actually needs, or external fragmentation when there is not enough contiguous memory available.
- Paging can reduce external fragmentation by ~~allowing unused~~ using available free memory efficiently, but can increase internal fragmentation by allowing unused portions of a page to be allocated.
- Segmentation can suffer from inefficiencies in memory usage due to fragmentation, limiting the number of processes that can be accommodated.

Overall, fragmentation can have a significant impact on memory allocation techniques in operating systems and can affect the efficiency and scalability of the system. It is important to consider the potential for fragmentation when selecting a memory allocation technique and to implement strategies to reduce or mitigate its effects.