UNIT 3 COMPLETE SOLVED

1	**Thrashing** A.is a natural consequence of virtual memory systems B.can always be avoided by swapping C.always occurs on large computers **D.**can be caused by poor paging algorithms E.None of the above
2	**Memory** A.is a device that performs a sequence of operations specified by instructions in memory. **B.**is the device where information is stored C.is a sequence of instructions D.is typically characterized by interactive processing and time-slicing of the CPU's time to allow quick response to each user. E.None of the above
3	**The principle of locality of reference justifies the use of** A.reenterable B.non reusable C.virtual memory **D.**cache memory E.None of the above
4	**Thrashing can be avoided if** **A.**the pages, belonging to the working set of the programs, are in main memory B.the speed of CPU is increased C.the speed of I/O processor is increased D.all of the above E.None of the above
5	**Fragmentation of the file system** A.occurs only if the file system is used improperly B.can always be prevented **C.**can be temporarily removed by compaction D.is a characteristic of all file systems E.None of the above
6	**The memory allocation scheme subject to "external" fragmentation is** **A.**segmentation B.swapping C.pure demand paging D.multiple contiguous fixed partitions E.None of the above
7	**Page stealing** A.is a sign of an efficient system **B.**is taking page frames from other working sets C.should be the tuning goal D.is taking larger disk spaces for pages paged out

	E. None of the above
8	**A page fault** A. is an error is a specific page B. occurs when a program accesses a page of memory **C.** is an access to a page not currently in memory D. is a reference to a page belonging to another program E. None of the above
9	**Which of the following statements is false?** A. a small page size causes large page tables **B.** internal fragmentation is increased with small pages C. a large page size causes instructions and data that will not be referenced brought into primary storage D. I/O transfers are more efficient with large pages E. None of the above
10	**An impulse turbine is used for** a. Low head of water b. High head of water c. Medium head of water d. High discharge
11	**The address of a page table in memory is pointed by:** a. stack pointer b. page table base register c. page register d. program counter
12	**What is compaction?** a. a technique for overcoming internal fragmentation b. a paging technique c. a technique for overcoming external fragmentation d. a technique for overcoming fatal error
13	**With relocation and limit registers, each logical address must be _______ the limit register.** a. Less than b. equal to c. greater than d. None of these
14	**The first fit, best fit and worst fit are strategies to select a ______.** a. Process from a queue to put in memory b. Processor to run the next process c. Free hole from a set of available holes d. All of these
15	**A system is in the safe state if:**

	a. The system can allocate resources to each process in some order and still avoid a deadlock b. There exist a safe sequence c. both (a) and (b) d. None of the mentioned
16	**Banker's algorithm for resource allocation deals with** A. deadlock prevention B. deadlock avoidance C. deadlock recovery D. mutual exclusion E. None of the above
17	**Which policy replace a page if it is not in the favored subset of a process's pages?** A. FIFO B. LRU C. LFU D. Working set E. None of the above

1	**What is the function of a lazy swapper?** In demand paging, a page is brought into the memory for its execution only when it is demanded. Demand Paging uses a lazy swapper to swap the pages from memory. An important feature of a lazy swapper is that it never swaps until that page is needed.
2	**What do you mean by thrashing? What is its cause?** If the process does not have number of frames it needs to support pages in active use, it will quickly page-fault. The high paging activity is called thrashing. In other words we can say that when page fault ratio decreases below level, it is called thrashing. Causes of Thrashing It result in severe performance problems. 1) If CPU utilization is too low then we increase the degree of multiprogramming by introducing a new process to the system. A global page replacement algorithm is used. The CPU scheduler sees the decreasing CPU utilization and increases the degree of multiprogramming. 2) CPU utilization is plotted against the degree of multiprogramming. 3) As the degree of multiprogramming increases, CPU utilization also increases. 4) If the degree of multiprogramming is increased further, thrashing sets in and CPU utilization drops sharply. 5) So, at this point, to increase CPU utilization and to stop thrashing, we must decrease the degree of multiprogramming.
3	**Distinguish between logical address space and physical address space.**

COMPARISON	LOGICAL ADDRESS	PHYSICAL ADDRESS
Basic	It is the virtual address generated by CPU	The physical address is a location in a memory unit.
Address Space	Set of all logical addresses generated by CPU in reference to a program is referred as Logical Address Space.	Set of all physical addresses mapped to the corresponding logical addresses is referred as Physical Address.
Visibility	The user can view the logical address of a program.	The user can never view physical address of program

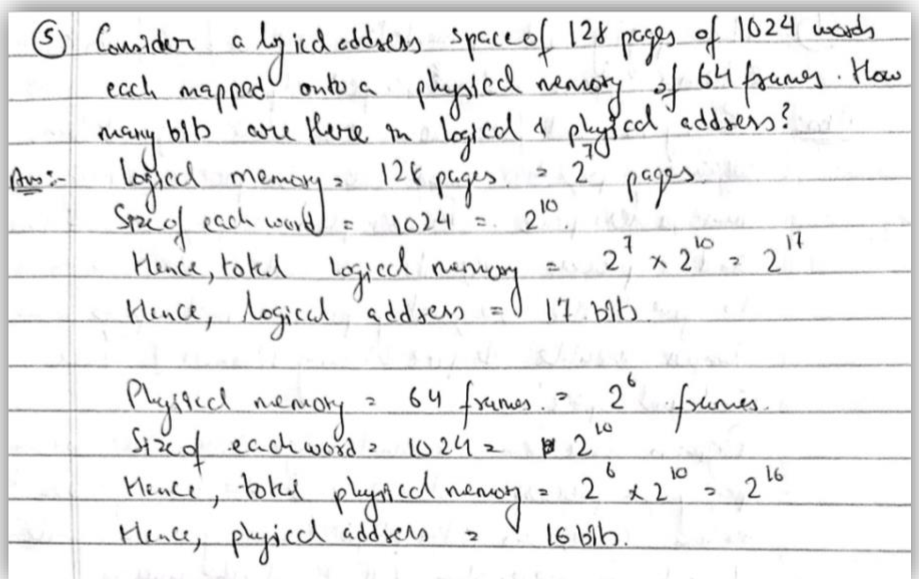
Access	The user uses the logical address to access the physical address.	The user can not directly access physical address.
Generation	The Logical Address is generated by the CPU	Physical Address is Computed by MMU

4 **What is the purpose of valid / invalid bit in demand paging?**

In demand paging, only the pages that are required currently are brought into the main memory.Assume that a process has 5 pages: **A, B, C, D, E** and **A** and **B** are in the memory. With the help of valid-invalid bit, the system can know, when required, that pages **C, D** and **E** are not in the memory.

In short: a 1 in valid-invalid bit signifies that the page is in memory and 0 signifies that the page may be invalid or haven't brought into the memory just yet.

5 **Consider a logical address space of 128 pages of 1024 words each mapped onto a physical memory of 64 frames. How many bits are there in logical and physical address?**


(5) Consider a logical address space of 128 pages of 1024 words each mapped onto a physical memory of 64 frames. How many bits are there in logical & physical address?

Ans:- logical memory = 128 pages = 2^7 pages.

Size of each word = 1024 = 2^{10}.

Hence, total logical memory = $2^7 \times 2^{10} = 2^{17}$

Hence, logical address = 17 bits.

Physical memory = 64 frames = 2^6 frames.

Size of each word = 1024 = 2^{10}

Hence, total physical memory = $2^6 \times 2^{10} = 2^{16}$

Hence, physical address = 16 bits.

6 **What is virtual memory?**

Virtual memory is a memory management technique where secondary memory can be used as if it were a part of the main memory. Virtual memory is a very common technique used in the operating systems (OS) of computers.

Virtual memory uses hardware and software to allow a computer to compensate for physical memory shortages, by temporarily transferring data from random access memory (RAM) to disk storage. In essence, virtual memory allows a computer to treat secondary memory as though it were the main memory.

7 **How paging and segmentation are not free from internal & external fragmentation respectively?**

	Paging are not free from internal fragmentation because a page has fixed size, but processes may request more or less space. Say a page is 32 units, and a process requests 20 units. Then when a page is given to the requesting process, that page is no longer useable despite having 12 units of free "internal" space. Paging are free from external fragmentation because in paging, a process is *allowed* to be allocated spaces that are non-contiguous in the physical memory. Meanwhile, the logical representation of those blocks will be contiguous in the virtual memory. Segmentation also suffers from external fragmentation as a segment of a process is laid out contiguously in physical memory and fragmentation would occur as segments of dead processes are replaced by segments of new processes. Segmentation, however, enables processes to share code; for instance, two different processes could share a code segment but have distinct data segments. Pure paging does not suffer from external fragmentation, but instead suffers from internal fragmentation. Processes are allocated in page granularity and if a page is not completely utilized, it results in internal fragmentation and a corresponding wastage of space. Paging also enables processes to share code at the granularity of pages
8	**What is the use of wait-for-graph?** Wait-for-graph is one of the methods for detecting the deadlock situation. This method is suitable for smaller database. In this method a graph is drawn based on the transaction and their lock on the resource. If the graph created has a closed loop or a cycle, then there is a deadlock.
9	**Define Priming in pump and why it is necessary?**
10	**Define slip and percentage of slip.**
11	**Differentiate between internal and external fragmentation.**

S.NO	INTERNAL FRAGMENTATION	EXTERNAL FRAGMENTATION
1.	In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	In external fragmentation, variable-sized memory blocks square measure appointed to method.
2.	Internal fragmentation happens when the method or process is larger than the memory.	External fragmentation happens when the method or process is removed.
3.	The solution of internal fragmentation is best-fit block.	Solution of external fragmentation is compaction, paging and segmentation.

	4.	Internal fragmentation occurs when memory is divided into fixed sized partitions.	External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.
	5.	The difference between memory allocated and required space or memory is called Internal fragmentation.	The unused spaces formed between non-contiguous memory fragments are too small to serve a new process, is called External fragmentation .
12	**How to recover from deadlock?** There are two options for breaking a deadlock: i) To abort one or more processes to break the circular wait (Process Termination). ii) To preempt some resources from one or more of deadlock processes (Resource Preemption).		
13	**Which parameters have to be maintained to implement the banker's algorithm?** Let n = number of processes, and m = number of resource types. – Available: Vector of length m indicates the number of available resources of each type. If available [j] = k, there are k instances of resource type Rj available. – Max: n x m matrix defines the maximum demand of each process. If Max [i,j] = k, then process Pi may request at most k instances of resource type Rj . – Allocation: n x m matrix defines the number of resources of each type currently allocated to each process. If Allocation[i,j] = k then Pi is currently allocated k instances of of resource type Rj. – Need: n x m matrix indicates the remaining resource need of each process. If Need[i,j] = k, then Pi may need k more instances of Rj to complete its task. Need [i,j] = Max[i,j] – Allocation [i,j].		
14	**What do you mean by swapping? Why do we need this?** A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution. Similar to round-robin CPU-scheduling algorithm , when a quantum expires, the memory manager will swap out that process to swap another process into the memory space that has been freed . Need of swapping: It is required when there is a need of better processing speed and the RAM is not enough to handle that. Although, there comes a point when even swapping does not work properly. Here is when one would require to enhance the RAM and everything would be back to normal.		
15	**What are the approaches we follow for the address binding?** A user program will go through several steps, before being executed as shown. Address binding of instructions and data to memory addresses can happen at three different stages – Compile time: If memory location known a priori, absolute code can be generated; must recompile code if starting location changes – Load time: Must generate relocatable code if memory location is not known at compile time		

	– Execution time: Binding delayed until run time if the process can be moved during its execution from one memory segment to another. Need hardware support for address maps (e.g., base and limit registers)
16	**What is the role of a page table in paging?** Page Table is a data structure used by the virtual memory system to store the mapping between logical addresses and physical addresses. Logical addresses are generated by the CPU for the pages of the processes therefore they are generally used by the processes. Physical addresses are the actual frame address of the memory. They are generally used by the hardware or more specifically by RAM subsystems.
17	**Differentiate between the basic and limit registers.** The base registers indicate where the page table starts in memory (this can be either a physical or logical addresses) and the limit register indicates the side of the table. The registers are usually not loaded directly. Their values are usually written to the hardware Process Context Block (PCB).
18	**What is the purpose of wait for graph? Justify your answer.** Wait-for-graph is one of the methods for detecting the deadlock situation. This method is suitable for smaller database. In this method a graph is drawn based on the transaction and their lock on the resource. If the graph created has a closed loop or a cycle, then there is a deadlock.
19	**What are the necessary conditions for deadlock?** Deadlock can arise if four conditions hold simultaneously • Mutual exclusion: only one process at a time can use a resource. If another process requests that resource, the requesting process must be delayed until the resource has been released. • Hold and wait: a process is holding at least one resource and is waiting to acquire additional resources held by other processes (i.e., you hold a resource and wait for another one). • No preemption: a resource can be released only voluntarily by the process holding it, after that process has completed its task (ex., in FCFS a process use the CPU until it terminates). • Circular wait: there exists a set {P0, P1, …, P0} of waiting processes such that P0 is waiting for a resource that is held by P1, P1 is waiting for a resource that is held by P2, …, Pn–1 is waiting for a resource that is held by Pn, and P0 is waiting for a resource that is held by P0.
20	**What is a resource-allocation graph?** The resource allocation graph is the pictorial representation of the state of a system. As its name suggests, the resource allocation graph is the complete information about all the processes which are holding some resources or waiting for some resources. It also contains the information about all the instances of all the resources whether they are available or being used by the processes.In Resource allocation graph, the process is represented by a Circle while the Resource is represented by a rectangle. Let's see the types of vertices and edges in detail.
21	**What is the basic approach of Page Replacement?** A page replacement algorithm looks at the limited information about accesses to the pages provided by hardware, and tries to guess which pages should be replaced to minimize the total number of page misses, while balancing this with the costs (primary storage and processor time) of the algorithm itself.

	If no frame is free is available, find one that is not presently being used and free it. A frame can be freed by writing its contents to swap space, and changing the page table to show that the page is no longer in memory. Now the freed frame can be used to hold the page for which the process faulted.
22	**What are the major problems to implement Demand Paging?** The two major problems to implement demand paging is developing a. Frame allocation algorithm b. Page replacement algorithm
23	**Give the basic concepts about paging.** Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.
24	**Explain the basic concepts of segmentation.** In Operating Systems, Segmentation is a memory management technique in which, the memory is divided into the variable size parts. Each part is known as segment which can be allocated to a process.The details about each segment are stored in a table called as segment table. Segment table is stored in one (or many) of the segments.
25	**Differentiate local and global page replacement algorithm.** Local replacement means that an incoming page is brought in only to the relevant process address space. Global replacement policy allows any page frame from any process to be replaced. The latter is applicable to variable partitions model only.
26	**How is memory protected in a paged environment?** Memory protection is a way to control memory access rights on a computer, and is a part of most modern instruction set architectures and operating systems. The main purpose of memory protection is to prevent a process from accessing memory that has not been allocated to it. Memory protection in a paged environment is accomplished by protection bits that are associated with each frame

PART C : LONG QUESTION ANSWERS

1	**Consider the following page reference string 1,2,3,4,5,3,4,1,6,7,8,7,8,1,7,6,2,5,4,5,3,2** **Calculate the number of page faults in each case using the following algorithms: (i) FIFO (ii) LRU (iii) Optimal. Assume the memory size is 4 frames.** NO ANSWERS FOUND.
2	**Write about segmentation with example discuss basic difference between paging and segmentation?** **Segmentation** Segmentation is a memory management technique in which each job is divided into several segments of different sizes, one for each module that contains pieces that perform related functions. Each segment is actually a different logical address space of the program. When a process is to be executed, its corresponding segmentation are loaded into non-contiguous memory though every segment is loaded into a contiguous block of available memory. Following are the important differences between Paging and Segmentation.

Sr. No.	Paging	Segmentation
1	In Paging, a process address space is broken into fixed sized blocks called pages.	In Segmentation, a process address space is broken in varying sized blocks called sections.
2	Operating System divides the memory into pages.	Compiler is responsible to calculate the segment size, the virtual address and actual address.
3	Page size is determined by available memory.	Section size is determined by the user.
4	Paging technique is faster in terms of memory access.	Segmentation is slower than paging.
5	Paging can cause internal fragmentation as some pages may go underutilized.	Segmentation can cause external fragmentation as some memory block may not be used at all.
6	During paging, a logical address is divided into page number and page offset.	During segmentation, a logical address is divided into section number and section offset.
7	During paging, a logical address is divided into page number and page offset.	During segmentation, a logical address is divided into section number and section offset.
8	Page table stores the page data.	Segmentation table stores the segmentation data.

3 **When do page fault occurs? Discuss the action taken by the operating system, when a page fault occurs.**

A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access, aborting the program if it is invalid. If it is valid a free frame is located and I/O requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.

Handling Page Fault.

1. Check the location of the referenced page in the PMT
2. If a page fault occured, call on the operating system to fix it
3. Using the frame replacement algorithm, find the frame location
4. Read the data from disk to memory
5. Update the page map table for the process
6. The instruction that caused the page fault is restarted when the process resumes execution.

4

Consider the following snapshot of a system.

	Allocation	**Max**	**Available**
	A B C D	**A B C D**	**A B C D**
P0	**0 0 1 2**	**0 0 1 2**	**1 5 2 0**
P1	**1 0 0 0**	**1 7 5 0**	
P2	**1 3 5 4**	**2 3 5 6**	
P3	**0 6 3 2**	**0 6 5 2**	
P4	**0 0 1 4**	**0 6 5 6**	

Using Banker's algorithm, answer the following questions.

(i) What is the content of matrix need?

(ii) Is the system in a safe state?

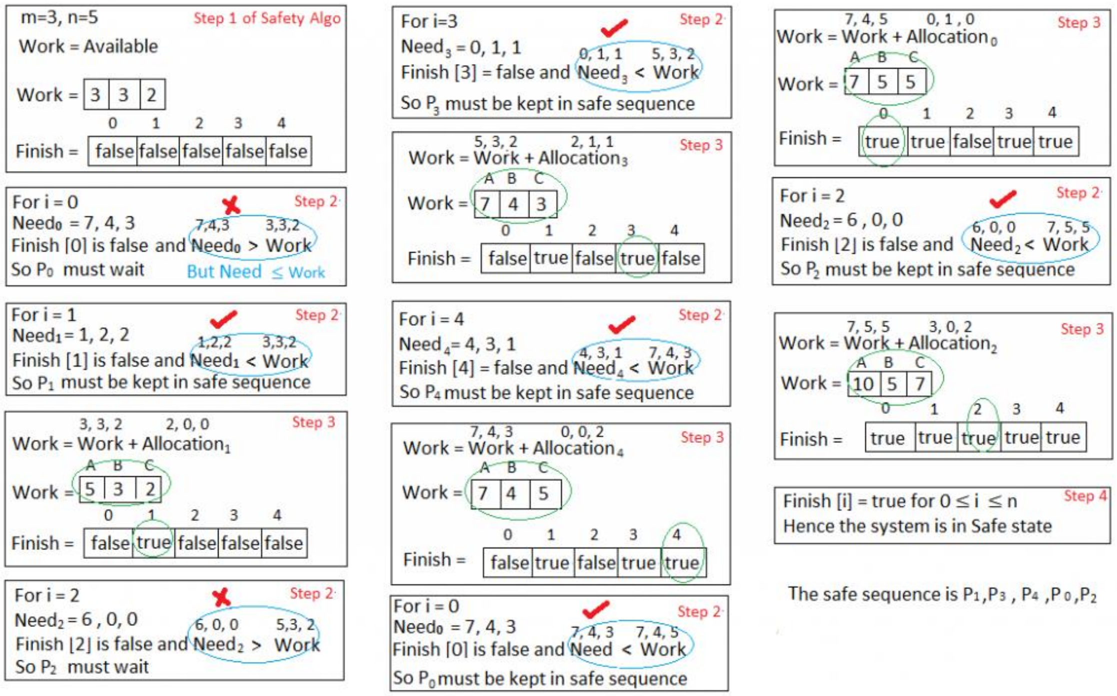
(iii) If a request from process P1 arrives for (0, 4, 2, 0) can the request be granted immediately?

(i) Need [i, j] = Max [i, j] – Allocation [i, j]

So, the content of Need Matrix is:

Process	Need		
	A	B	C
P_0	7	4	3
P_1	1	2	2
P_2	6	0	0
P_3	0	1	1
P_4	4	3	1

(ii) Yes, the system is safe. The safe sequence is P1, P3, P4, P0, P2.



(iii) The values **of** Need for **processes** P0 through P4 respectively are (**0**, **0**, **0**, **0**), (**0**, 7, 5, **0**), (1, **0**, **0**, **2**), (**0**, **0**, **2**, **0**), and (**0**, 6, **4**, **2**).

5

Distinguish between internal and external fragmentation. Provide any two solutions to avoid external fragmentation.

S.NO	INTERNAL FRAGMENTATION	EXTERNAL FRAGMENTATION
1.	In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	In external fragmentation, variable-sized memory blocks square measure appointed to method.
2.	Internal fragmentation happens when the method or process is larger than the memory.	External fragmentation happens when the method or process is removed.

3.	The solution of internal fragmentation is best-fit block.	Solution of external fragmentation is compaction, paging and segmentation.
4.	Internal fragmentation occurs when memory is divided into fixed sized partitions.	External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.
5.	The difference between memory allocated and required space or memory is called Internal fragmentation.	The unused spaces formed between non-contiguous memory fragments are too small to serve a new process, is called External fragmentation .

Solution to **external fragmentation**:
1) Compaction: shuffling **the fragmented** memory into one contiguous location.
2) Virtual memory addressing by using paging and segmentation.

6

Explain about the necessary conditions for deadlock. How we can prevent deadlock by using these?

When a process requests for the resource that is been held another process which needs another resource to continue, but is been held by the first process, then it is called a **deadlock**.

There are 4 conditions necessary for the occurrence of a deadlock.

1. **Mutual Exclusion:**
   When two people meet in the landings, they can't just walk through because there is space only for one person. This condition to allow only one person (or process) to use the step between them (or the resource) is the first condition necessary for the occurrence of the deadlock.
2. **Hold and Wait:**
   When the 2 people refuses to retreat and hold their grounds, it is called holding. This is the next necessary condition for the the deadlock.
3. **No Preemption:**
   For resolving the deadlock one can simply cancel one of the processes for other to continue. But Operating System doesn't do so. It allocates the resources to the processors for as much time needed until the task is completed. Hence, there is no temporary reallocation of the resources. It is third condition for deadlock.
4. **Circular Wait:**
   When the two people refuses to retreat and wait for each other to retreat, so that they can complete their task, it is called circular wait. It is the last condition for the deadlock to occur.

	Deadlocks can be avoided by avoiding at least one of the four conditions, because all this four conditions are required simultaneously to cause deadlock. **Mutual Exclusion** Resources shared such as read-only files do not lead to deadlocks but resources, such as printers and tape drives, requires exclusive access by a single process. **Hold and Wait** In this condition processes must be prevented from holding one or more resources while simultaneously waiting for one or more others. **No Preemption** Preemption of process resource allocations can avoid the condition of deadlocks, where ever possible. **Circular Wait** Circular wait can be avoided if we number all resources, and require that processes request resources only in strictly increasing(or decreasing) order.
7	**Write short notes on** **i. TLB** **ii. Semaphore** **(i)TLB** • Operating system divides each incoming programs into pages of equal size. the sections of the disks are called as blocks or sector. • The section of main memory are called as page frame, one sector will hold one page of job instruction and fit into one page of memory. • Fixed size blocks in main memory are called page frame and breaking of logical memory into blocks of same size are called pages. • CPU generates, logical address containing page number and an effect. the page number is used to retrieve the frame number from a page table which gives the page addr so that physical address is calculated as base + offset. • To implement the page table in memory set of register are used of page table size increases their register are not suitable, then special purpose hardware cache is used for page table called as translation look aside buffer (TLB) • TLB is very effective in improving the performance of the page frame access. the cache badtn is implemented in a technology which is faster then the primary memory technology. **(ii)Semaphore** **Semaphores** are often used to restrict the number of threads than can access some (physical or logical) resource.

	Semaphores are devices used to help with synchronization. If multiple processes share a common resource, they need a way to be able to use that resource without disrupting each other. You want each process to be able to read from and write to that resource uninterrupted. A semaphore will either **allow or disallow access to the resource**, depending on how it is set up. One example setup would be a semaphore which allowed any number of processes to read from the resource, but only one could ever be in the process of writing to that resource at a time. Semaphores are commonly use for two purposes: to **share a common memory space and to share access to files**. Semaphores are one of the techniques for inter process communication (IPC). The C programming language provides a set of interfaces or "functions" for managing semaphores.
8	**Given memory partitions of 120K, 520K, 320K, 324K and 620K (in order). How would each of the First fit, Best fit and worst fit algorithms place processes of 227K, 432K, 127K and 441K (in order)? Which algorithm makes the most efficient use of memory?** **NO ANSWERS FOUND**
9	**Distinguish between internal and external fragmentation. Provide any two solutions to avoid external fragmentation.** REFER QUESTION NO. 5
10	**Consider the following snapshot of a system Available Allocation Max** **A B C A B C A B C** **P0 0 1 0 7 5 3 3 3 2** **P1 2 0 0 3 2 2** **P2 3 0 2 9 0 2** **P3 2 1 1 2 2 2** **P4 0 0 2 4 3 3** **i. What is the content of the matrix Need?** **ii. Is the system in a safe state? If yes, what is the safe sequence? Show the detailed steps as per Banker's algorithm?** **REFER QUESTION NO. 4**
11	**Consider the following segment table** **Segment Base Length** **0 240 500** **1 2150 28** **2 180 60**

	3 1175 470 **4 1482 55** **What are the physical addresses for the following logical addresses?** **(a) 0,280 (b) 1,20 (c) 2,150 (d) 3,320 (e) 4,188** **NO ANSWERS FOUND**
12	**What do you mean by Thrashing? What are the methods to avoid Thrashing?** If the process does not have number of frames it needs to support pages in active use, it will quickly page-fault. The high paging activity is called thrashing. We must provide a process with as many frames as it needs. Several techniques are used. **The Working of Set Model (Strategy)** It starts by looking at how many frames a process is actually using. This defines the locality model. **Locality Model** It states that as a process executes, it moves from locality to locality. A **locality** is a set of pages that are actively used together. A program is generally composed of several different localities which overlap.