1)

First Fit:

In this, the first partition large enough to be assigned is given to the process.
The memory allocation for first fit is given below:

Partition's (MB)	100	170	40	205	300	185
Process Size (MB)	15	75		200	185	175

As you can see above there is no space for a process to have a size of 80 MB.

Best Fit:

In this, the partition is greater if not equal to the process.
The memory allocation for best fit is given below:

Partition's (MB)	100	170	40	205	300	185
Process Size (MB)	75	80	15	200	175	185

As you can see all the process were allocated in this approach.

Worst Fit:

In this the largest partition is assigned to the process.
The memory allocation for worst fit is given below:

Partition's (MB)	100	170	40	205	300	185
Process Size (MB)	80	75		15	200	185

As you can see above there is no space for a process to have a size of 175 MB

In these tests we have clearly determined that best fit is indeed the best fit for the multiprogramming fixed number of tasks.

4)

a) Given, virtual address space is 216 bytes.

Physical address space = 216 bytes

Page size is 4096 bytes which would be22 bytes

Total pages in page table=216/212 = 24 or 16 pages. That is 0 to 15 in decimal and 0 to F in hexadecimal.

The page size and the frame size is the same.

Similarly, number of frames are also 16 from 0 to 15 in decimal and 0 to F in hexadecimal.

Given, virtual address Ox621C

Last 12 bits represent page offset and leftmost 4 bits represent page number. Here,

i) Ox621C --> page no. 6 present at frame 8.

Physical address will be Ox821C.

ii) OxF0A3--> page no. F or 15 in decimal, is present at frame 2,
Physical address will be Ox20A3.

iii) OxBC1A--> page B or 11 present at frame 4.

Physical address will be Ox4C1A.

iv) Ox5BAA--> page 5 present at frame 13 or D in physical address.

Physical address will be Ox DBAA

v) Ox0BA1--> page 0 is present at frame 9.

Physical address will be Ox9BA1.

Page reference 6,15,11,5,0 will be set to 1.

b) Page numbers 1, 9, and 14 are not present in physical frames or main memory, according to the chart, because frame entries in these pages are empty.

As a result, logical addresses Ox1AF0, Ox9A12, OxE432 and others will cause a page fault.

c) Only page frames 3, 14, and 11 are unoccupied in main memory, hence page faults can be met using LRU from these three frames.

5) "ma" stands for memory access time. Effective access time=memory access time as long as there are no page faults. We must read the page from the disc and then access the desired word if a page failure occurs. Let p denote the likelihood of a page fault (0<=p<=1). (1-p)*ma+p*page fault time is the effective access time.

effective access time = (0.8) (1 microsec)+ (0.1) (2 microsec) +(0.1) (5002 microsec)
= 501.2 microsec

= 0.5 millisec

6) Thrashing is a term used in computer science to describe when a computer's virtual memory subsystem is constantly paging, swapping data in memory for data on disc, to the exclusion of most application-level activity. This causes the computer's performance to deteriorate or perhaps fail. The problem could last continuously unless the root cause is addressed.

Thrashing in virtual memory systems can be caused by programmes or workloads with insufficient locality of reference: if a program's or workload's working set can't be successfully held within physical memory, constant data swapping, or thrashing, can happen. The name was coined to describe the sound tapes made when data was rapidly written to and read from them during the tape operating system's early days. Many older low-end computers lack enough RAM (memory) to support modern usage patterns and adding more memory can often make the device run substantially faster. This gain in performance is due to the decreased quantity of paging required.

A user can use any of the following to stop thrashing caused by excessive paging:

- Replace memory-intensive programmes with less memory-intensive alternatives.
- Reduce the number of programmes running on your computer.
- Increase the computer's RAM capacity.
- Assign programme working priority, such as low, normal, and high.

8) DMA boosts system concurrency by allowing the CPU to work on other tasks while the DMA system transports data between the system and memory buses. Because the DMA controller must be incorporated into the system and the system must allow the DMA controller to be a bus master, hardware design is challenging. Cycle stealing may also be required to share the memory bus between the CPU and the DMA controller.