Q1

Assume that the size of the computer memory is 100KB, and there are 7 processes that need to use the memory with sizes of 20KB, 30KB, 10KB, 40KB, 15KB, 35KB, and 25KB, respectively. The processes enter the system in sequence. Write a program to simulate the arrival of processes and the allocation of memory using fixed partition and variable partition algorithms, and **output the starting address of each memory block allocated to each process and the final memory free space.**

**Requirements:**

1. The fixed partition algorithm divides the memory into 4 partitions with sizes of 15KB, 20KB, 30KB, and 35KB.
2. Both partition algorithm uses the best-fit algorithm for memory allocation, that is, it finds the smallest available memory block that can satisfy the size of the process.
3. The processes enter the system in sequence.
4. The process structure includes the process ID, size.
5. The starting address of the memory partition is 0, and the maximum memory block that a process can allocate is 100KB, and the size of the allocated memory block is in KB.
6. The program needs to output the starting address of each memory block allocated to each process and the final memory free space.
7. When a new process arrives and the memory is full, an executed process needs to be selected to release its memory space, and the new process is allocated to that memory space. The selection rule is to choose the process with the smallest available size (Assuming the previous process has completed when the current process arrives.).
8. C language should be used to implement the simulation program.

Q2

Write a ANSI C program to fulfill the following operating system tasks in demand paging main memory management:

Assume we have a 460-word program, with the following sequence of requests for program words: 10, 11, 104, 170, 73, 309, 185, 245, 246, 434, 458 and 364. The page frame size should match the size of the individual pages into which the program has been divided. Namely, the number of page frames in the main memory, is **the total number divided by the page size**. For instance, if the main memory is 200 and the page size is 100, the number of page frames in the main memory would be two. Then the requests 10 and 11 are located on Page 0, and requests 104 and 170 are on Page 1.

**Requirements:**

1. Develop a C language program that simulates a 460-word program with the given request sequence.
2. The program should allow for two configurable parameters: **main memory size** and **page size**.
3. Implement both First-In, First-Out (FIFO) and Least Recently Used (LRU) page replacement algorithms.
4. For each algorithm, calculate the hit rate for the request sequence under two case:
   1. Assume the memory size is 200 words, page size is 100 words (there are two page frames).
   2. Memory: 200 words, page size: 20 words (10 pages, 0 through 9).