DESIGN DOC

Design decisions:

- Values of p, q, n, m, t are taken from the user, along with time of execution and which placement strategy to use.

- Memory is simulated using an array, where value ‘0’ at an index signifies that the memory located is unallocated and value ‘1’ at an index signifies that the memory location is allocated to a process.

- Every second, r threads are created to simulate process arrival rate ‘r’ per sec. These threads then, in their thread function, make memory allocation request, by adding their data to a queue.

- A specific thread is created with the sole purpose of allocating memory according to process requests. This thread basically checks whether the request corresponding to the front element of the queue can be fulfilled or not. If it can be, memory is allocated and the queue is dequeued, else the thread sleeps for 1 sec and then repeats the cycle.

- A partcular process(thread) is made aware of the fact that it has been allocated memory, by changing the value at a memory location from ‘-1’ to start index of allocated memory.

- After a thread is allocated memory, sleep function is used to simulate the duration of the process and finally memory deallocation function is called to deallocate the memory.

- Mutex lock is used to ensure that process allocation and deallocation do not happen at the same time, but at a particular time, only one of them runs.

- At the end of ‘T’, that is, when time exceeds the time of execution, the program ends by displaying memory allocation percentage and average turnaround time.
  Memory allocation percentage is calculated by how many indexes have value ‘1’ divided by array size.
  Average turnaround number is calculated by noting time taken by proceeses to get requested memory (only for those processes which were successfully allocated memory) divided by total number of such processes.