Problem 1 (0 Points)
Consider a dynamically partitioned memory that has two holes with sizes: 1300 KB and 1200 KB. Assume that memory requests arrive in the following order: A) 1000 KB, B) 1100 KB, and C) 250 KB. Compare the results of the four allocation algorithms first-fit, best-fit, next-fit, and worst-fit.

a) Use a table to show the evolution of free space as each memory request is handled. The table should have five columns (memory request, and free space sizes for the four algorithms) and four rows (initial free space, and each memory request).

b) For the above workload, rank the four algorithms by memory utilization. By speed.

c) Do the results for the above workload correspond to expectations for general workloads? Explain.

Problem 2 (0 Points)
Consider a buddy system and the address 011011110000.

a) If the block size associated with this address is 8 bytes, what is the binary address of the buddy?

b) What is the largest block size \( N \) such that the above address still has a buddy? Explain.

Problem 3 (4 Points)
Consider a buddy system and the address 100100001000. Assume the largest possible \( U \) and smallest possible \( L \).

a) If the block size is 8 bytes, what is the binary address of the buddy?

b) What is the largest block size \( N \) such that the above address still has a buddy? Explain.

c) Let \( b_k(x) \) be the buddy of address \( x \) with block size \( 2^k \). Write an expression (NOT an algorithm) for \( b_k(x) \). Explain why the form of the expression is correct.

d) Demonstrate that your expression in Part c is correct.

Problem 4 (0 Points)
If page table entries are 4 bytes each and the page size is 8 KB, how many levels of page tables would be required to map a 32-bit address space if the top level page table fits into a single page?
Problem 5 (2 Points)
Suppose that a logical address space has eight pages where each page is 1,024 bytes and this logical address space is mapped onto a physical memory of 32 page frames.

a) How many bits are there in the logical address?
b) How many bits are there in the physical address?
c) Describe the structure of the page table for this system.

Problem 6 (6 Points)
Suppose that a process has been allocated 4 page frames (numbered 0 to 3) and initially (time = 0), frame k contains virtual page k (e.g., frame 2 contains page 2). Furthermore, suppose that the following is the reference string (RS) where the memory operation is indicated by a 'r' (read) or 'w' (write):

2r 0w 3r 1w 4r 1r 0w 1r 2r 3r

For the page replacement algorithms below, fill out the table below which shows the evolution of the page table:

<table>
<thead>
<tr>
<th>Time T</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>2r</td>
<td>0w</td>
<td>3r</td>
<td>1w</td>
<td>4r</td>
<td>1r</td>
<td>0w</td>
<td>1r</td>
<td>2r</td>
<td>3r</td>
<td></td>
</tr>
</tbody>
</table>

Assume that initially all R-bits and M-bits are 0 and that the clock cursor is pointing at frame 0.

a) Optimal; b) FIFO; c) LRU; and d) Enhanced Clock (M-bit version).

Problem 7 (6 Points)
Stallings (Fifth Edition), Problem 8.8. Use the following notation for the machine code:

- \( R_n \leftarrow x \): Load Register \( n \) from location \( X \)
- \( x \leftarrow R_n \): Store Register \( n \) to location \( X \)
- \( R_n \leftarrow R_a + R_b \): Add Registers \( R_a \) and \( R_b \) storing the result in register \( R_n \) (Other arithmetic operators have the same form)
- \( \text{go to } L \): Go to label \( L \)
- \( \text{if } (R_a < R_b) \text{ goto } L \): Test and branch to label \( L \) (Other logic operators have the same form)

Make up your own notation that matches the above for any other instructions that you may require.