Assignment 6
Deadlocks and Virtual Memory Management
CSCI 5103, Fall 2016
Due November 15, 2016
This assignment must be done individually.

Problem 1: (15 points) Consider a system with consisting of \( m \) resources of the same type being shared by \( n \) processes. Prove that the system is deadlock free if:
1. \( \text{Need}(i) > 0 \) for \( i = 1, 2, 3, \ldots, n \), and
2. The sum of all maximum needs is less than \( m+n \).

Problem 2: (15 points) We can obtain the banker’s algorithm for a single resource type from the general banker’s algorithm simply by reducing the dimensionality of the various arrays by 1. Show through an example that the multiple-resource-type banker’s algorithm cannot be implemented by individual application of the single-resource-type scheme to each resource type.

Problem 3: (15 points) A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows;

<table>
<thead>
<tr>
<th></th>
<th>Allocated</th>
<th>Maximum</th>
<th>Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process A</td>
<td>1 0 2 1 1</td>
<td>1 1 2 1 3</td>
<td>0 0 x 1 2</td>
</tr>
<tr>
<td>Process B</td>
<td>2 0 1 1 0</td>
<td>2 2 2 1 0</td>
<td></td>
</tr>
<tr>
<td>Process C</td>
<td>1 1 0 1 0</td>
<td>2 1 3 1 0</td>
<td></td>
</tr>
<tr>
<td>Process D</td>
<td>1 1 1 1 0</td>
<td>1 1 2 2 1</td>
<td></td>
</tr>
</tbody>
</table>

What is the smallest value of \( x \) for which this is a safe state?

Problem 4: (20 points) Consider the two-dimensional array \( A \):

```java
int A[][] = new int[100][100];
```

where \( A[0][0] \) is at location 200, in a paged system with pages of size 200. A small process is in page 0 (locations 0 to 199) for manipulating the matrix; thus, every instruction fetch will be from page 0.
For three page frames, how many page faults are generated by the following array-initialization loops, using LRU replacement, and assuming page frame 1 has the process in it, and the other two are initially empty:

```java
for (int j = 0; j < 100; j++)
  for (int i = 0; i < 100; i++)
    A[i][j] = 0;
for (int i = 0; i < 100; i++)
  for (int j = 0; j < 100; j++)
    A[i][j] = 0;
```
**Problem 5: (15 points)** A computer has four page-frames. The time of loading, time of last access, and the $R$ and $M$ bits for each page are as shown below (the times are in clock ticks).

<table>
<thead>
<tr>
<th>Page</th>
<th>Load time</th>
<th>Access time</th>
<th>R</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>126</td>
<td>280</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>230</td>
<td>265</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>140</td>
<td>270</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>110</td>
<td>285</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(a) Which page will NRU replace?  
(b) Which page will LRU replace?  
(c) Which page will second chance replace?  
(d) Which page will FIFO replace?

**Problem 6: (10 points)** Suppose that a machine has 38-bit virtual addresses and 32-bit physical addresses.

(a) What is the main advantage of a multilevel page table over a single-level one?  
(b) With a two-level page table, 16-KB pages and 4-byt entries, how many bits should be allocated for the top-level page table field and how many for the second-level page table field? Explain.

**Problem 7: (10 points)** A small computer has 4 page frames. At the first clock tick, the $R$ bits are 0111 (page 0 is 0 and the others are 1). At subsequent clock ticks, the values are 1011, 1000, 1100, 0001, 0010, 1010, and 0100. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last tick.  
If a page-fault occurs after the 8th clock tick, which page will be evicted?