Problem 1. What is the difference between deadlock and starvation? (5 Points)

Problem 2. Assume that the pseudo-codes for two threads of a process were given as the following frames. Give the smallest and largest number of registers r1, r2, and r3 that could hold the value “2” after this code runs (and explain how). Do we need to consider the race condition or process synchronization issue?

<table>
<thead>
<tr>
<th>%thread 1</th>
<th>%thread 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1 = a;</td>
<td>r3 = b;</td>
</tr>
<tr>
<td>r2 = a;</td>
<td>a = r3;</td>
</tr>
<tr>
<td>if (r1==r2)b = 2;</td>
<td></td>
</tr>
</tbody>
</table>

Problem 3. P_A, P_B and P_C are three programs in a file system. Assume that the pseudo codes for program P_A be given as the following:

```c
main() {
    n = 3;
    for(i=0; i<5; i++) {
        if (mod(n,2) == 0)
            statements
        else callSub()
        n = mod(n*17, 5);
    }
}
```

```c
CallSub() {
    Statements
}
```

Initially, a=0 and b=1. Assume that the for-loop is contained by page #4, the reminder of the main() is covered by page #5, and callSub() is contained by page #8. Assume that the first instruction on page #8 forks a new process which executing P_B. Assume that the page reference string is made by recording the virtual page number every 2 ticks as the process is executing. When P_B is executed, the page reference string shown below is recorded:

```
{4, 5, 4, 5, 6, 6, 6}.
```

Assume that the first instruction on page #5 forks a new process which executing P_C. P_C will have the page reference string shown below:

```
{7, 7, 7, 8}.
```

There are two processes in the system at t=0: P_1 executes P_A, and P_2 executes P_B. Assume that the OS needs 3 ticks to handle each page fault, and page faults are handled on a first-come-first-serve basis. Draw a Gantt chart to illustrate the scheduling of these processes and their child processes and to compute the waiting times and number of page faults for each process if a Round-Robin non-preemptive schedule with a time quantum of 5 ticks is used. Note that the issue of page replacement is ignored. Show your work.
Consider the Deadlock Avoidance problem. Suppose we have four types of resources: 10 of type-A, 5 of type-B, 7 of type-C, and 8 of type-D. In the current state, maximum resources are needed, and the resources allocated to each of the processes are given as the following:

<table>
<thead>
<tr>
<th>Process</th>
<th>Max</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>9 5 5 5</td>
<td>2 0 2 1</td>
</tr>
<tr>
<td>P2</td>
<td>2 2 3 3</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>P3</td>
<td>7 5 4 4</td>
<td>4 1 0 2</td>
</tr>
<tr>
<td>P4</td>
<td>3 3 2 2</td>
<td>1 1 0 0</td>
</tr>
<tr>
<td>P5</td>
<td>4 4 4 4</td>
<td>1 0 1 1</td>
</tr>
</tbody>
</table>

a) Calculate the need matrix and the available array.

b) Show that the current state is safe, that is, show a safe sequence of processes. (Show your work.)

c) Given the request (3, 2, 3, 3) from P5. Should this request be granted? (Show your work.)