1. For each of the following statements, decide TRUE or FALSE. **You MUST justify your answer to receive full credit.** (10%)  
(1) Shared memory, semaphores, pipes, and signals are all inter-process communication (IPC) mechanisms. (2%)  
(2) A process is in trashing state if it is blocked and cannot proceed. (2%)  
(3) In multiprogramming, it is safe to have multiple threads/processes reading the same data. (2%)  
(4) Deadlock cannot happen with the use of semaphores. (2%)  
(5) With virtual memory, the kernel must be invoked to resolve every memory reference. (2%)  

2. Consider the following snapshot of a system:  

<table>
<thead>
<tr>
<th>Resource</th>
<th>Allocated</th>
<th>Max Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W X Y Z</td>
<td>W X Y Z</td>
</tr>
<tr>
<td>Process ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2 0 1 2</td>
<td>3 5 3 6</td>
</tr>
<tr>
<td>2</td>
<td>2 3 0 1</td>
<td>5 4 3 2</td>
</tr>
<tr>
<td>3</td>
<td>1 2 2 0</td>
<td>3 4 3 2</td>
</tr>
<tr>
<td>4</td>
<td>1 1 2 1</td>
<td>5 3 3 3</td>
</tr>
</tbody>
</table>

Currently available resource \([W, X, Y, Z] = [2, 2, 2, 2]\). Will the system be in a safe state if Process 4’s request of \([0, 0, 1, 0]\) is granted at this moment? **You must show step-by-step how you arrive at your answer to receive full credit.** (9%)  

3. The following processes are submitted to an operating system. All processes are CPU-bound without input/output. Assume that a time quantum is 2, where applicable. A lower priority number indicates a greater priority.  

<table>
<thead>
<tr>
<th>process ID</th>
<th>arrival time</th>
<th>burst time</th>
<th>priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Match the following scheduling methods and schedules. Fill in each **blank** with the letter (A thru H) of the corresponding scheduling method. Each letter is used at most once. (16%)
Scheduling methods:
A. shortest job first (SJF)
B. shortest remaining time first (SRTF)
C. round robin (RR)
D. first come first served (FCFS)
E. priority (non-preemptive)
F. least recently used (LRU)
G. priority (preemptive)
H. none of the above

Schedules:
(1) (4%)

<table>
<thead>
<tr>
<th>time</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>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

(2) (4%)

<table>
<thead>
<tr>
<th>time</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>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

(3) (4%)

<table>
<thead>
<tr>
<th>time</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>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(4) (4%)

<table>
<thead>
<tr>
<th>time</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>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>proc</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

4. The following figure shows the pipelined datapath with control signals. For the instruction sequence below, please answer the following questions. (15%)

```
add $1, $2, $3
and $1, $1, $4
lw $5, 4($1)
sub $7, $5, $6
or $8, $5, $7
```
國立臺灣科技大學 104 學年度碩士班招生試題
系所組別：資訊工程系碩士班
科 目：資訊工程概論

(總分為 100 分)
1. Please give the control signals of multiplexors A, B, C and control lines D, E, F for clock cycle 5. Note that the answer for each signal should be one of the follows: 0, 1, 00, 01, 10, or 11. (6%)

<table>
<thead>
<tr>
<th>Control Signal</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. For clock cycle 5, what are sent through lines X, Y, and Z? Please write down the corresponding register numbers ($1$, $2$ … $8$). (4%)

<table>
<thead>
<tr>
<th>Register Number</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. How many cycles does it take to finish the execution of this instruction sequence? (5%)

5. Below are some assumptions of memory access times:
   - 1 memory bus clock cycle to send the address
   - 10 memory bus clock cycles for each DRAM access initiated
   - 1 memory bus clock cycle to send a word of data

Based on these assumptions, please calculate the miss penalty of a four-word block for the following organizations of memory. (10%)

1. One-word-wide memory organization: the memory and the bus between the processor and the memory are both one word wide. (3%)
2. Wider memory organization: the memory and the bus between the processor and the memory are widened to two words. (3%)
3. Interleaved memory organization: the memory chips are organized in four banks, where each bank is one word wide, without widening the interconnection bus (in other words, the bus is still one word wide). (4%)

6. Consider the following performance measurements for a program:

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Computer A</th>
<th>Computer B</th>
<th>Computer C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction count</td>
<td>10 billion</td>
<td>10 billion</td>
<td>9 billion</td>
</tr>
<tr>
<td>Clock rate</td>
<td>2 GHz</td>
<td>4 GHz</td>
<td>2 GHz</td>
</tr>
<tr>
<td>CPI</td>
<td>1.1</td>
<td>2.4</td>
<td>1.2</td>
</tr>
</tbody>
</table>

For each of the following statements, decide TRUE or FALSE. (10%)

1. Computer B is faster than Computer A for this program. (2%)
2. Computer C is faster than Computer A for this program. (2%)
3. Computer C has higher MIPS (million instructions per second) rating than
Computer A for this program. (2%)  
(4) Suppose the program spends 10% of time on addition and 60% of time on  
multiplication. This program can run 2 times faster by improving only the  
speed of addition. (2%)  
(5) Suppose the program spends 10% of time on addition and 60% of time on  
multiplication. This program can run 2 times faster by improving only the  
speed of multiplication. (2%)  

7. Describe the running time of the following pseudocode fragments using Big-Oh  
notation in terms of the variable n. In using Big-Oh, make the most precise statement  
about the running time that’s possible. Assume all variables used have been declared.  
(15%)  
(1) (5%)  
int compute_value(int k) {  
    int value;  
    for (int i = 0; i < k; ++i)  
        value = value + (i * k);  
    return value;  
}  
  
int s;  
for (int i = 0; i < n; ++i) {  
    if (n < 1000)  
        s++;  
    else  
        s += compute_value (n);  
}  

(2) (5%)  
for (int j = 4; j < n; j=j+2) {  
    val = 0;  
    for (int i = 0; i < j; ++i)  
        val = val + i * j;  
    for (int k = 0; k < n; ++k) {  
        val++;  
    }  
}  

(3) (5%)
for (int i = 0; i < n * 1000; ++i) {
    sum = (sum * sum)/(n * i);
    for (int j = 0; j < i; ++j) {
        sum += j * i;
    }
}

8. Consider the following function:
int test(int n) {
    int s;
    if (n > 0) {
        s = (test(n-1) + test(n/3) + 15);
        return s;
    } else
        return 0;
}

Write down the complete recurrence relation, T(n), for the running time of test(n). Please remember to include a base case T(0). Note that solving the relation is not required, just write it down. (5%)

9. Find the minimal spanning tree for the following connected weighted graph G in the order that Kruskal’s algorithm includes them.
B-D __________________________ (10%)