Exercise 3.1
Consider the following program (as machine codes represented by C for the clarity):

```c
int foo(int* a){
    boolean weight = false;
    int b = 0;
    for (i=0; i< 10; i++) {
        if (a[i] >= 10 && weight == false) {
            weight = true;
            b += a[i]*2;
            compute some things based on b;
        } else {
            b += a[i];
            compute some others;
            compute some things based on b;
        }
    }
    return b;
}
```

1. Draw the control flow graph (CFG) of the above program, and define the basic blocks such that there is no branch within each basic block.

2. Write the integer linear programming to get the worst-case execution time (WCET) of the above program by assuming that you have already known the worst-case execution time in each basic block.
Exercise 3.2
Worst-Case Execution Time (WCET)

(a) Why is it not a good idea, in order to simplify the problem, to assume that all memory accesses result in cache misses while performing the WCET analysis?

(b) Is it possible to derive the WCET based on the source code? If yes, what are the required conditions and assumptions?

(c) Is it, in general, safe to analyze the WCET for different components of the micro-architecture (like caches and CPU cycles) separately, and then add up the individual results to compute the total WCET?

(d) Mr. Smart has derived the WCET 4 ms of a program on a (deterministic) platform with 1GHz CPU frequency. The system somehow is too powerful and can be slowed down to run with 500MHz CPU frequency (but all the rest of the platform remain the same) to reduce the power consumption. Mr. Smart concludes that the WCET under this new setting is 8 ms for this program. Is that a safe estimation? Is the estimation too pessimistic? If it is pessimistic, how can he improve the estimation? If it is not pessimistic, why not?

Exercise 3.3
Consider the following program:

```c
read a; 
read b; 
read a; 
if (a>b) { 
  read c; 
  read d; 
} else { 
  read e; 
  read f; 
} 
read x; 
read a; 
```

(a) Perform a May and a Must-Analysis on this program, assuming an LRU-cache with associativity 4 that is empty at the start of the program. Is it possible to determine whether the last access to “a” results in a cache hit or a cache miss?

(b) We now assume that the cache uses the FIFO replacement policy instead. Could an analysis determine whether the last access to “a” results in a cache hit or a cache miss if the cache is empty at the start of the program?
Exercise 3.4
Consider the following case with two sporadic tasks and 1 server (to be defined later):

<table>
<thead>
<tr>
<th></th>
<th>$\tau_1$</th>
<th>$\tau_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>$T_i$</td>
<td>2</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_s$</td>
<td>1.6</td>
</tr>
<tr>
<td>$T_s$</td>
<td>4</td>
</tr>
</tbody>
</table>

Can we guarantee that task $\tau_2$ always meets its deadline in the following settings? Explain your answer. If the answer is “no”, what is the maximum capacity ($C_s$) of the server to ensure that $\tau_2$ can meet its deadline.
(a) setting the above server to a constant bandwidth server (CBS) and using EDF
(b) setting the above server to a polling server and using RM
(c) setting the above server to a deferrable server and using RM
(d) setting the above server to a sporadic server and using RM

Exercise 3.5
When designing the constant bandwidth server (CBS), we have the following description:

- When the server is idle at time $t$ and a job arrives, if $t < D_{Si}$ and $\frac{b_{Si}}{D_{Si}} - t < \frac{C_{Si}}{T_{Si}}$, the server becomes active with the same budget and server deadline; otherwise, $D_{Si}$ is set to $t + T_{Si}$ and $b_{Si}$ is set to $C_{Si}$.

What happens if we change the description to the following statement?

- When the server is idle at time $t$ and a job arrives, if $t < D_{Si}$, the server becomes active with the same budget and server deadline; otherwise, $D_{Si}$ is set to $t + T_{Si}$ and $b_{Si}$ is set to $C_{Si}$.

Exercise 3.6
Mr. Ing is asked to provide his customer Carl quality of service guarantee with respect to his working hours. Carl has no idea about the arrival of the requests, but would like to request Mr. Ing to provide 12.5% of his working time to work for these requests. For serving each request, Mr. Ing requires at most 15 minutes. Carl also requires Mr. Ing to reply a request within the following $x$ working hours (the time that Mr. Ing is not on duty is not counted) as a hard guarantee if there is no pended request from Carl. The payment for such a contract is proportional to $1/x$.

Assume that Mr. Ing works 8 hours a day and has his own scheduling policy for his working duties. Please explain how Mr. Ing can ensure that the service guarantee to Carl is always fulfilled and maximize the profit of this contract.

- Case 1: Mr. Ing does not have any other obligated duties
- Case 2: Mr. Ing has other obligated duties that are periodic