ItIV HW1

Problem 1

1.

Iteration	LHS Q_0	B_0	RHS	Stop
1	30	30	30	Yes

$R_0 = Q_0 + C_0 = 30 + 10 = 40$

2.

Iteration	LHS Q_1	B_1	j	$Q_1 + \tau$	T_j	$\lceil \frac{Q_1+\tau}{T_j} \rceil$	C_j	RHS	Stop
1	30	30	0	30.1	50	1	10	40	No
2	40	30	0	40.1	50	1	10	40	Yes

$R_1 = Q_1 + C_1 = 40 + 30 = 70$

3.

Iteration	LHS Q_2	B_2	j	$Q_2 + \tau$	T_j	$\lceil \frac{Q_2+\tau}{T_j} \rceil$	C_j	RHS	Stop
1	20	20	0, 1	20.1	50, 200	1, 1	10, 30	60	No
2	60	20	0, 1	60.1	50, 200	2, 1	10, 30	70	No
3	70	20	0, 1	70.1	50, 200	2, 1	10, 30	70	Yes

$R_2 = Q_2 + C_2 = 70 + 20 = 90$

Problem 2

1.

```
1.440000
2.040000
2.560000
3.160000
3.680000
4.280000
5.200000
8.400000
9.000000
9.680000
10.200000
19.360000
19.800000
20.320000
29.400000
29.760000
30.280000
```

2.

The MATLAB code of a CAN bus timing analysis is as follows:

```
filename = "input.dat";

fildID = fopen(filename, 'r');

n_task = str2num(fgetl(fildID));
tau = str2num(fgetl(fildID));
data = fscanf(fildID, "%f %f %f\n", [3 Inf]);
priority = data(1,:);
time = data(2,:);
period = data(3,:);

response_time = [];
for i_task = 1 : n_task
    block_time = max(time(i_task:end));
    Q = block_time;
    while true
        if(i_task == 1) RHS = block_time;
        else RHS = block_time + sum(ceil((Q+tau)./period(1:i_task-1)).*time(1:i_task-1));
        end
        if(Q == RHS) response_time(i_task) = Q + time(i_task); break; end
        Q = RHS;
    end
end

fprintf("%f\n",response_time)
```

Problem 3

1.

Iteration	LHS R_0	C_0	RHS	Stop
1	10	10	10	Yes

$R_0 = 10$

2.

Iteration	LHS R_1	C_1	j	R_1	T_j	$\lceil \frac{R_1}{T_j} \rceil$	C_j	RHS	Stop
1	30	30	0	30	50	1	10	40	No
2	40	30	0	40	50	1	10	40	Yes

$R_1 = 40$

3.

Iteration	LHS R_2	C_2	j	R_2	T_j	$\lceil \frac{R_2}{T_j} \rceil$	C_j	RHS	Stop
1	20	20	0, 1	20	50, 200	1, 1	10, 30	60	No
2	60	20	0, 1	60	50, 200	2, 1	10, 30	70	No
3	70	20	0, 1	70	50, 200	2, 1	10, 30	70	Yes

$R_2 = 70$

4.
If a lower-priority task is currently transmitting on the CAN, a higher-prioiry task might be blocked in a non-preemptive scheme, resulting in higher response time. On the other hand, all lower-priority tasks won't affect a higher-priority-task trasmmision under a preemptive scheme, which can achieve lower response time.

Problem 4

1.
The duplicated schedule pattern: (4, 10, 1, 2, 6, 7)
2.
The frame arrival pattern: (4, 10, 0, 3, 5, 6, 10, 13, 15, 16)
3.
The schedule pattern: (4, 10, 1, 2, 6, 7, 11, 12, 16, 17)
4.

k	$\max_{1\le j\le n}(s_{j+k} - s_j)$	=	$\min_{1\le j\le m}(a_{i+k-1} - a_j)$	=	
1	$\max_{1\le j\le n}(s_{j+1} - s_j)$	4	$\min_{1\le j\le m}(a_i - a_j)$	0	4
2	$\max_{1\le j\le n}(s_{j+2} - s_j)$	5	$\min_{1\le j\le m}(a_{i+1} - a_j)$	1	4
3	$\max_{1\le j\le n}(s_{j+3} - s_j)$	9	$\min_{1\le j\le m}(a_{i+2} - a_j)$	3	6
4	$\max_{1\le j\le n}(s_{j+4} - s_j)$	10	$\min_{1\le j\le m}(a_{i+3} - a_j)$	6	4

5.
The worst-case response time is $\max(4, 4, 6, 4) + 1 = 7$.