



**2.** (5 points) Does the RM policy in the above problem lead to any missed deadlines? How much time should be shown in the timing diagram to give one confidence that the diagram is “proof” of how the system works?

**3.** (5 points) For the above problem, compute the utilization,

$$U := \sum_{i=1}^3 \frac{C_i}{T_i} \tag{1}$$

Recall that if this quantity is less than or equal to  $N(2^{1/N} - 1)$  (where  $N$  is the number of tasks), then the RM policy is guaranteed not to miss deadlines. Since  $N = 3$  in this case,  $N(2^{1/N} - 1) \approx 0.78$ . Is  $U \leq 0.78$ ? If not, are you still sure that there are no missed deadlines? Explain.

4. (5 points) You are working at a fast-paced new start-up in IoT technology. Your team has written up some RTOS code (not written in FreeRTOS, but in a similar OS), and have asked you to complete the task function (see below). You are to use the API functions:

```
void LED_on(int LEDNumber)
```

```
void vdelay(int delay_time)
```

```
void LED_off(int LEDNumber)
```

The task function should turn the LED on, delay for 100 ms, turn the LED off, and then delay for another 100 ms. You can assume that the delay function parameter is in units of milliseconds. To the best of your abilities, provide a complete C-language implementation for **vLEDTask**. Use the page following the program listing:

```
#include <stdlib.h>
```

```
#include <RTOS.h>
```

```
#include <board.h>
```

```
/* global variables */
```

```
int redval=0;
```

```
int greenval=1;
```

```
int blueval=2;
```

```
void vLEDTask(void *pvParameters)
```

```
{
```

```
    int colour = *((int *) pvParameters);
```

```
    /* please write the missing code! */
```

```
}
```

```
/*
```

```
    * main() function
```

```
*/
```

```
int main(void) {
```

```
    BoardSetup();
```

```
    /* create tasks */
```

```
    xTaskCreate(vLEDTask, "RedTask", &redval, idle_priority_level+3);
```

```
    xTaskCreate(vLEDTask, "GreenTask", &greenval, idle_priority_level+2);
```

```
    xTaskCreate(vLEDTask, "BlueTask", &blueval, idle_priority_level+1);
```

```
    vTaskStartScheduler();
```

```
    /* should never arrive here */
```

```
    return 1;
```

```
}
```

Your solution: