# DVA 438 – Group 5 Test Cases

To run these tests cases, include their respective header in your main file (i.e. "*grp5_testcase1*") and then call the respective test case start function (i.e. *start_testCase1()*). In addition, in the test case itself you have to change the "`#include "kernel/ROSA_Extended.h"`" to your respective API header.

## Test Case 1

Checks dynamic task creation/deletion and if it there is any issues with deleting a task which holds a semaphore. This is done by having one task create another which instantly takes a semaphore the moment it gets to run, then delays indefinitely. The parent task then deletes the newly created task which holds the semaphore, it then tries to take the semaphore to see if it has been released. This behavior is observable by LED0 which will be turned on/off every second if this works correctly.

In addition to this, there is also two lower (same) priority tasks which runs concurrently with the rest of the test. These tasks continuously turn on/off LED2 and LED3 in order to see that there is no starvation in the system.

### Expected Output (required to pass the test case):

- LED0 (the first on the board) should turn on/off every second.
- LED2 and LED3 should be lit up.

## Test Case 2

Tests if a call to a delay function on same priority tasks using the same semaphore can cause issues in the queues: i.e. if T1 takes sem1 and delays, T2 then tries to take sem1 (gets denied), T1 is woken up but put behind T2 in the queue if using FIFO queue. Now T2 waits for T1 to give up the semaphore, but T1 will never run as it is behind T2 in the ready queue, thus creating a deadlock. This case is implemented and observable through LED0 and LED1 which will take turns being lit up each second if the OS handles this issue correctly.

In addition to this, there is also two lower (same) priority tasks which runs concurrently with the rest of the test. These tasks continuously turn on/off LED2 and LED3 in order to see that there is no starvation in the system.

### Expected Output (required to pass the test case):

- LED0 and LED1 takes turn being lit up each second.
- LED2 and LED3 should be lit up.

## Test Case 3

Checks if task suspension is handled correctly in regards to waiting tasks, i.e. if a task gets suspended and resumed when it's in the waiting queue, it should return to the waiting queue and not the ready queue. This is implemented by having two tasks turning on/off their respective LEDs on different timing intervals, and the task with the lower period suspending and resuming the other task to see if it is put back into the waiting queue. If the OS handles this issue, then the test will turn LED0 on/off each second, and LED1 will be turned on/off every half a second.

In addition to this, there is also two lower (same) priority tasks which runs concurrently with the rest of the test. These tasks continuously turn on/off LED2 and LED3 in order to see that there is no starvation in the system.

## Expected Output (required to pass the test case):

- LED0 is turned on/off each second.
- LED1 is turned on/off each half a second.
- The fifth LED on the board does not turn red (LED4).
- LED2 and LED3 should be lit up.