Round-Robin Task Scheduling

Part A

Here's my truncated output

```
xenomai@ieu:~/exercises/round_robin$ ./ex05a
start task  : 0
Task  : 0
start task  : 1
Task  : 1
start task  : 2
Task  : 2
wake up all tasks
Running Task  : 0  at time ms : 10
.
.
.
Running Task  : 0  at time ms : 200
End Task  : 0
Running Task  : 1  at time ms : 10
.
.
.
Running Task  : 1  at time ms : 200
End Task  : 1
Running Task  : 2  at time ms : 10
.
.
.
Running Task  : 2  at time ms : 200
End Task  : 2
```

Tasks execute based on when they are started as they are all the same priority. Since task 0 is started first with task 1 next and task 2 following, they start in this order.

Part B

I set the time credit alloted for task 0, 1, and 2 to be 1e5 nanoseconds. The output I then got was:

```
xenomai@ieu:~/exercises/round_robin$ ./ex05b
start task  : 0
Task  : 0
start task  : 1
Task  : 1
start task  : 2
Task  : 2
wake up all tasks
Running Task  : 0  at time ms : 10
Running Task  : 1  at time ms : 10
Running Task  : 2  at time ms : 10
Running Task  : 0  at time ms : 20
Running Task  : 1  at time ms : 20
Running Task  : 2  at time ms : 20
Running Task  : 0  at time ms : 30
Running Task  : 1  at time ms : 30
Running Task  : 2  at time ms : 30
Running Task  : 0  at time ms : 40
Running Task  : 1  at time ms : 40
Running Task  : 2  at time ms : 40
Running Task  : 0  at time ms : 50
Running Task  : 0  at time ms : 60
Running Task  : 1  at time ms : 50
Running Task  : 2  at time ms : 50
Running Task  : 0  at time ms : 70
Running Task  : 1  at time ms : 60
Running Task  : 2  at time ms : 60
Running Task  : 0  at time ms : 80
Running Task  : 1  at time ms : 70
Running Task  : 2  at time ms : 70
Running Task  : 0  at time ms : 90
Running Task  : 1  at time ms : 80
Running Task  : 2  at time ms : 80
Running Task  : 0  at time ms : 100
Running Task  : 1  at time ms : 90
Running Task  : 2  at time ms : 90
Running Task  : 0  at time ms : 110
Running Task  : 1  at time ms : 100
Running Task  : 2  at time ms : 100
Running Task  : 0  at time ms : 120
```

```
Running Task  : 1  at time ms : 110
Running Task  : 0  at time ms : 130
Running Task  : 1  at time ms : 120
Running Task  : 2  at time ms : 110
Running Task  : 0  at time ms : 140
Running Task  : 1  at time ms : 130
Running Task  : 2  at time ms : 120
Running Task  : 1  at time ms : 140
Running Task  : 2  at time ms : 130
Running Task  : 0  at time ms : 150
Running Task  : 1  at time ms : 150
Running Task  : 2  at time ms : 140
Running Task  : 0  at time ms : 160
Running Task  : 1  at time ms : 160
Running Task  : 2  at time ms : 150
Running Task  : 0  at time ms : 170
Running Task  : 1  at time ms : 170
Running Task  : 2  at time ms : 160
Running Task  : 0  at time ms : 180
Running Task  : 1  at time ms : 180
Running Task  : 2  at time ms : 170
Running Task  : 0  at time ms : 190
Running Task  : 1  at time ms : 190
Running Task  : 2  at time ms : 180
Running Task  : 0  at time ms : 200
End Task  : 0
Running Task  : 1  at time ms : 200
End Task  : 1
Running Task  : 2  at time ms : 190
Running Task  : 2  at time ms : 200
End Task  : 2

Type CTRL-C to end this program
```

You can see that in the given time slice of 1e5 nanoseconds, each task can only execute one spin time cycle before round robin scheduling switches what task is executing. Since task 0, 1, and 2 are started in this order, the pattern above is seen.

Part C

Here's my output:

```
xenomai@ieu:~/exercises/round_robin$ ./ex05c
start task  : 0
Task  : 0
start task  : 1
Task  : 1
start task  : 2
Task  : 2
start task  : 3
Task  : 3
wake up all tasks
Running Task  : 3  at time ms : 10
Running Task  : 3  at time ms : 20
Running Task  : 3  at time ms : 30
Running Task  : 3  at time ms : 40
Running Task  : 3  at time ms : 50
Running Task  : 3  at time ms : 60
Running Task  : 3  at time ms : 70
Running Task  : 3  at time ms : 80
Running Task  : 3  at time ms : 90
Running Task  : 3  at time ms : 100
Running Task  : 3  at time ms : 110
Running Task  : 3  at time ms : 120
Running Task  : 3  at time ms : 130
Running Task  : 3  at time ms : 140
Running Task  : 3  at time ms : 150
Running Task  : 3  at time ms : 160
Running Task  : 3  at time ms : 170
Running Task  : 3  at time ms : 180
Running Task  : 3  at time ms : 190
Running Task  : 3  at time ms : 200
End Task  : 3
Running Task  : 0  at time ms : 10
Running Task  : 1  at time ms : 10
Running Task  : 2  at time ms : 10
Running Task  : 0  at time ms : 20
Running Task  : 1  at time ms : 20
Running Task  : 2  at time ms : 20
Running Task  : 0  at time ms : 30
Running Task  : 1  at time ms : 30
Running Task  : 2  at time ms : 30
Running Task  : 0  at time ms : 40
```

```
Running Task  : 1  at time ms : 40
Running Task  : 2  at time ms : 40
Running Task  : 0  at time ms : 50
Running Task  : 1  at time ms : 50
Running Task  : 2  at time ms : 50
Running Task  : 0  at time ms : 60
Running Task  : 1  at time ms : 60
Running Task  : 2  at time ms : 60
Running Task  : 0  at time ms : 70
Running Task  : 1  at time ms : 70
Running Task  : 2  at time ms : 70
Running Task  : 0  at time ms : 80
Running Task  : 1  at time ms : 80
Running Task  : 2  at time ms : 80
Running Task  : 0  at time ms : 90
Running Task  : 1  at time ms : 90
Running Task  : 2  at time ms : 90
Running Task  : 0  at time ms : 100
Running Task  : 1  at time ms : 100
Running Task  : 2  at time ms : 100
Running Task  : 0  at time ms : 110
Running Task  : 1  at time ms : 110
Running Task  : 2  at time ms : 110
Running Task  : 0  at time ms : 120
Running Task  : 1  at time ms : 120
Running Task  : 2  at time ms : 120
Running Task  : 0  at time ms : 130
Running Task  : 1  at time ms : 130
Running Task  : 2  at time ms : 130
Running Task  : 0  at time ms : 140
Running Task  : 1  at time ms : 140
Running Task  : 2  at time ms : 140
Running Task  : 0  at time ms : 150
Running Task  : 1  at time ms : 150
Running Task  : 2  at time ms : 150
Running Task  : 0  at time ms : 160
Running Task  : 1  at time ms : 160
Running Task  : 2  at time ms : 160
Running Task  : 0  at time ms : 170
Running Task  : 1  at time ms : 170
Running Task  : 2  at time ms : 170
Running Task  : 0  at time ms : 180
Running Task  : 1  at time ms : 180
Running Task  : 2  at time ms : 180
```

```
Running Task  : 0  at time ms : 190
Running Task  : 1  at time ms : 190
Running Task  : 2  at time ms : 190
Running Task  : 0  at time ms : 200
End Task  : 0
Running Task  : 1  at time ms : 200
End Task  : 1
Running Task  : 2  at time ms : 200
End Task  : 2

Type CTRL-C to end this program
```

By having the fourth task (task 3) have a greater priority then task 0, 1, and 2, priority based scheduling takes over. Task 3 begins to execute first and completes before the other tasks execute. It does not matter than task 3 is given an allotted time credit and included in round robin scheduling because of its higher priority.