# Coursera's
# Development of Real-Time Systems
# Peer-graded Assignment: Assignment 1

## Requirement

Create task functions containing the following functionality

Task1 should print out "This is task 1" every 100 milliseconds (hint use fflush(stdout) after printf())

Task2 should print out "This is task 2" every 500 milliseconds

### Task Source Code

```
int main( void )
{
    printf("Hello");
    /* This demo uses heap_5.c, so start by defining
    is only used for test and example reasons.  Heap
    http://www.freertos.org/a00111.html for an expla
    prvInitialiseHeap();

    main_task();

    /* Do not include trace code when performing a c

    return 0;
}
```

1- Creat Tasks in main_task function

```
TaskHandle_t myTaskHandle =NULL ;

#define TASK1_STACKSIZE  1000
#define TASK2_STACKSIZE  100
#define TASK1_PRIORITY   3
#define TASK2_PRIORITY   1


void main_task( void )
{


    xTaskCreate(Task1_100ms, "task1", TASK1_STACKSIZE, NULL, TASK1_PRIORITY , &myTaskHandle);
    xTaskCreate(Task2_500ms, "task2", TASK2_STACKSIZE, NULL, TASK2_PRIORITY , &myTaskHandle);

    vTaskStartScheduler();

    while(1)
    {

    }
}
```

2- Tasks implementation

```c
void Task1_100ms()
{
    while(1)
    {
        printf("This is task 1 \n");

        vTaskDelay(100);  /* 100ms */
    }
}


void Task2_500ms()
{
    while(1)
    {
        printf("This is task 2 \n");

        vTaskDelay(500);  /* 500ms */
    }
}
```

## The Output of the Task

```
Hello
Trace started.
The trace will be dumped to disk if a call to configASSERT() fails.
Uncomment the call to kbhit() in this file to also dump trace with a key press.
This is task 1
This is task 2
This is task 1
This is task 1
This is task 1
This is task 1
This is task 1
This is task 2
This is task 1
This is task 1
This is task 1
This is task 1
This is task 1
This is task 1
This is task 2
This is task 1
This is task 1
This is task 1
This is task 1
This is task 1
This is task 2
```