```
Test Preview: Summary for jsh114 of c4
--------------------------------------

  Public Tests:
    Marks for coding exercises::  12 / 12

Git Repo: git@gitlab.doc.ic.ac.uk:lab1819_autumn/496-bayesian-regression_jsh114.git
Commit ID: 0c676
```

```
# -*- coding: utf-8 -*-

"""
Use this file for your answers.

This file should been in the root of the repository
(do not move it or change the file name)

"""

import numpy as np

def lml(alpha, beta, Phi, Y):
    """
    4 marks

    :param alpha: float
    :param beta: float
    :param Phi: array of shape (N, M)
    :param Y: array of shape (N, 1)
    :return: the log marginal likelihood, a scalar
    """

    N, M = Phi.shape
    bI = beta * np.eye(N)

    term = alpha*Phi.dot(Phi.T) + bI
    term_1 = -0.5 * N * np.log(2*np.pi)
    term_2 = -0.5 * np.log(np.linalg.det(term))
    term_3 = -0.5 * Y.T.dot(np.linalg.inv(term)).dot(Y)

    return np.asscalar(term_1 + term_2 + term_3)

def grad_lml(alpha, beta, Phi, Y):
    """
    8 marks (4 for each component)

    :param alpha: float
    :param beta: float
    :param Phi: array of shape (N, M)
    :param Y: array of shape (N, 1)
    :return: array of shape (2,). The components of this array are the gradients
    (d_lml_d_alpha, d_lml_d_beta), the gradients of lml with respect to alpha and ⤤
beta respectively.
    """
    N, M = Phi.shape
    bI = beta * np.eye(N)
    f = alpha*Phi.dot(Phi.T) + bI
    detF = np.linalg.det(f)
    invF = np.linalg.inv(f)

    dF_da = Phi.dot(Phi.T)
    dF_db = np.eye(N)

    dlml_da = np.asscalar(
        float(-0.5*detF*np.trace(invF.dot(dF_da)))/detF +
        -0.5*Y.T.dot(-invF.dot(dF_da).dot(invF)).dot(Y)
    )

    dlml_db = np.asscalar(
        float(-0.5*detF*np.trace(invF.dot(dF_db)))/detF +
        -0.5*Y.T.dot(-invF.dot(dF_db).dot(invF)).dot(Y)
    )

    return np.array([dlml_da, dlml_db])

```

```
1: -------- Test Output --------
2:
3:
4: -------- Test Errors --------
5: ..
6: ----------------------------------------------------------------------
7: Ran 2 tests in 0.001s
8:
9: OK
```