Answers to question 1 and 2:

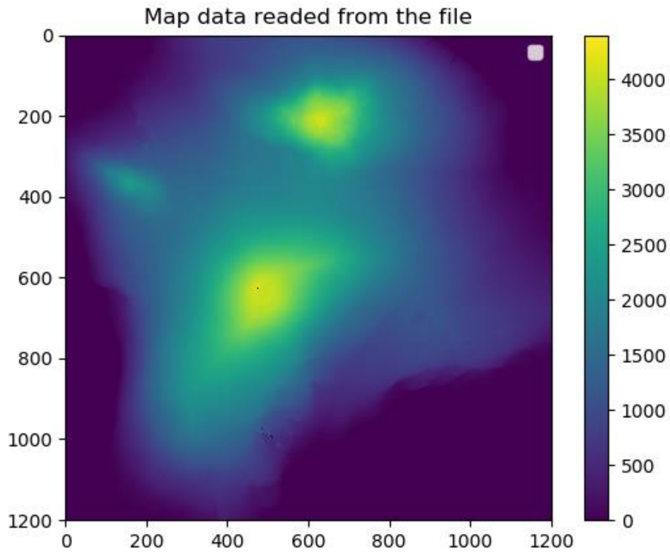
Question 1 a:

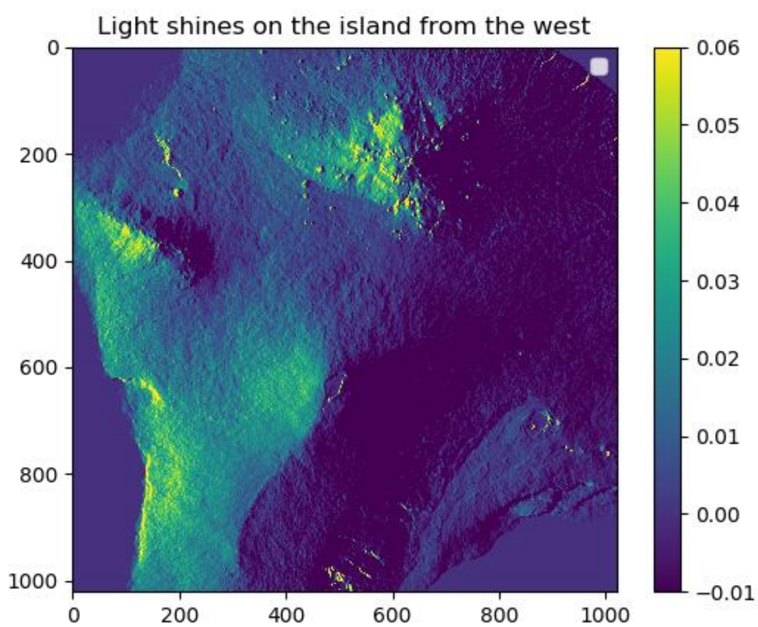
Question1(a) Pseudo code

```
def pesudo_for_1a():
        array[1021][1021]                                    //initalize a 2D array with size 1021 x 1021
        file = file.open(Map data)
        for y from 0 to 1021:
                for x from 0 to 1021:
                        read map data into array;
        //when x, y == 0, calculate gradient using foreward difference, when x,y == 1021, use back difference;
        //use central difference other cases;
        gradient_map = gradient(array)
        I = calculate I using gradient
        plot the graph of w
        plot the graph of I

def gradient(2D_array):
          for x from 0 to 1021:
                        for y from 0 tov1021:
                                if x == 0 or x == 1020:
                                        if x == 0:
                                                partial_x = derivitive_forward(2D_array,x,y)
                                        else:
                                                partial_x = derivitive_backward(2D_array,x,y)
                                else:
                                        partial_x = derivitive_central(2D_array,x,y)
                                if y == 0 or y == 1020:
                                        if y == 0:
                                                partial_y = derivitive_forward(2D_array,x,y)
                                        else:
                                                partial_y = derivitive_backward(2D_array,x,y)
                                else:
                                        partial_y = derivitive_central(2D_array,x,y)
                                map_data[x][y] = (partial_x, partial_y)
  return map_data
```

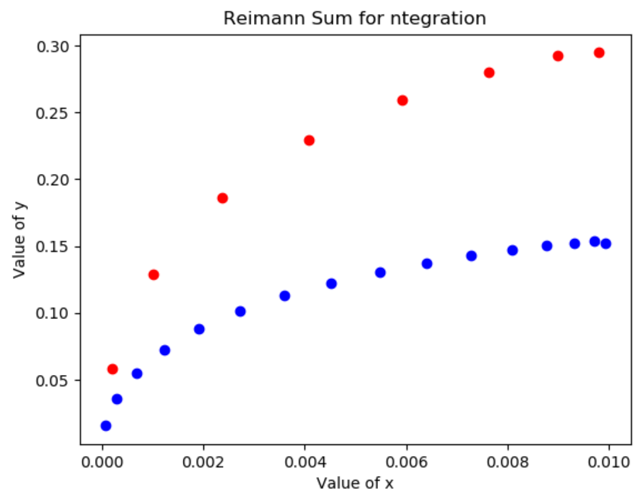
Question 1b:





Question 2a:

The code for this question is implement as method *Questrion_2a* Question_1_and_2.py, compare to the classical approximation of the period, the result approximate with N = 8 has a relative (fractional) error of around 4% and the result approximate with N = 16 has a relative error of around 2%. After plotting $w_k g_k$ on the graph, we discover that approximation with N = 8 have a lager value of each of the point than approximation with N = 16.



Although the value for N = 8 is generally greater than the value of N = 16, but since there are more data points, therefore the value for N = 16 are greater, or in another words, the result will be more accurate.

Question 2b:

For classical particle on spring, the energy is conserved. For the particle traveling at light speed when x = 0. The energy of the particle is:

$$E = \frac{1}{2} \times 1 \times (3.0 \times 10^8)^2$$

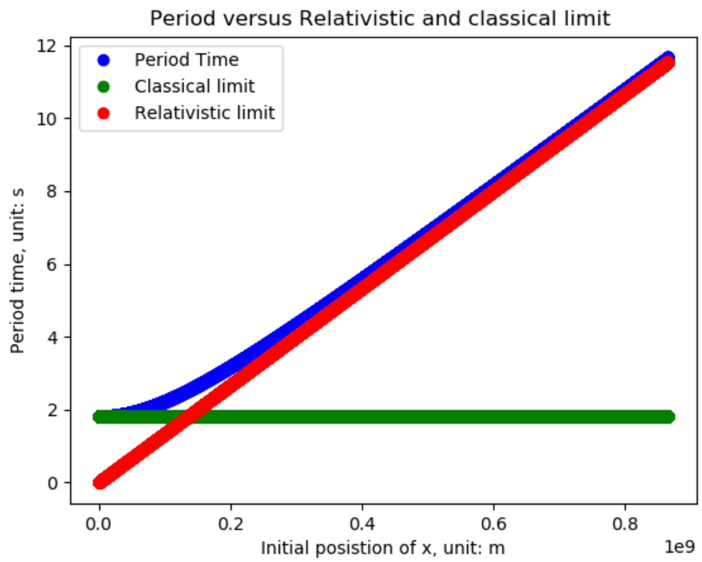
and since the energy is conserved, the initial displacement is:

$$T = \sqrt{\frac{2E}{k}} \approx 86602504.4m$$

Therefore, $x_c = 86602504.4m$.

Question 2c:

For small angle cases with N = 200, the estimation of error for this small angel case (0.01m) is 0.003528065523162205 calculated by equation (8) in the lab handout. This error corresponds to a relative error less than 0.2%. The plot of T between 1m < x < 10x_c, are shown below:



The graph showed that the as the initial position of x becomes greater and greater, the oscillation period of the particle on the spring is indeed closer to the relativistic limit of $4x_0/c$. When the initial position x_0 is small, the period of oscillation is close to the classical limit.