Exercise 2 (for submitting)

1. Investigate the performance of the steepest descent algorithm (gradient method) for the function

$$f(x, y) = 100 * (y - x^2)^2 + (1 - x)^2.$$

(a) Write a code for the steepest decent algorithm, as detailed below.
R users: define the function

```
sdescent = function( a0, tol, maxiter),
```

where $a0$ is an initial point (an array of two elements, corresponding to x and y), `tol` is tolerance and `maxiter` is the maximum number of iterations. The function will output a list which contains the following two values:

- 'a' : the solution (array of x and y).
- 'num': number of iterations.

Python: define

```
def sdescent(a0,tol,maxiter)
```

with the same input defined above. Output should be a list which contains the solution and number of iterations.
Stopping rule. The algorithm should stop if at least one of the following two conditions is met:

- Number of iterations reaches `maxiter`.
- It holds that $\|a_t - a_{t-1}\|_2 < tol$ (t is iteration number), i.e the norm of the difference between current and previous iterates solution is smaller than `tol`.

(b) Plot the function and the points produced by the algorithm.
(c) What can you say about the speed of convergence?

2. Investigate the performance of the Newton method for the function

$$f(x, y) = 100 * (y - x^2)^2 + (1 - x)^2.$$

(a) Write a code for the Newton method. R users: define the function

```
newton = function( a0, tol, maxiter),
```

Python users: define

```
def newton(a0,tol,maxiter).
```

Input, output and stopping rule are identical to that of question 1(a).
(b) Plot the function and the points produced by the algorithm.
(c) What can you say about the speed of convergence?