OpenCL exercise 2: Mandelbrot Set

1 Syntax

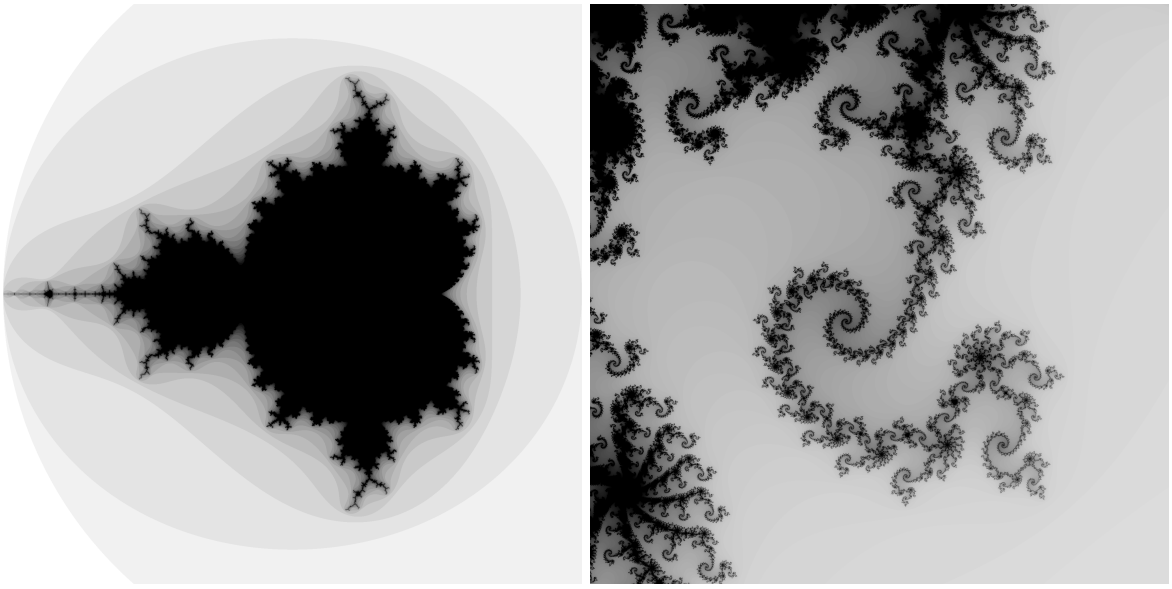
On the host:

```
cl::CommandQueue::enqueueNDRangeKernel(cl::Kernel kernel,
    cl::NDRange offset, cl::NDRange global, cl::NDRange local,
    eventsToWaitFor = NULL, cl::Event* event=NULL) const;
queue.enqueueNDRangeKernel(kernel, cl::NullRange,
    cl::NDRange(globalX, globalY), cl::NDRange(localX, localY), ...);
```

On the device:

```
size_t i = get_global_id(0); // Get global index of the current work item in the x-direction
size_t j = get_global_id(1); // Get global index of the current work item in the y-direction
```

2 Mandelbrot set



Consider the complex sequence defined by $z_0 = 0$ and $z_{n+1} = z_n^2 + c$ with $c \in \mathbb{C}$, $z \in \mathbb{C}$ and $n \in \mathbb{N}$. The Mandelbrot set is the set of numbers c such that the sequence z_n remains bounded for all $n \in \mathbb{N}$. It can be shown that the sequence diverges if $|z_n| > 2$ for any n. This property is useful for representing an approximation of the Mandelbrot set: The number n of iterations until $|z_n| > 2$ for a grid of values c is used to color the complex plane.

2.1 Task 1

Implement the calculation of the mandelbrot set on the GPU. Use a 2D index space, with one work item for each pixel of the result image. Add code for showing the time needed by the CPU, by the GPU and by the memory transfer and calculate the speedup.

Check the results.

2.2 Task 2

Use the second parameter set. Explain the results and the effect on the speedup.

2.3 Task 3

Modify the number of work items per work group and the number of work groups. What happens?