Exercise 3

Part 1

Use Amdahl's Law to answer the following questions:

a) Assume your program has a method *M* which cannot be parallelized, and this method accounts for *40%* of the total program runtime. What is the limit of the Speedup of your program when running on a machine with *n* processors.
b) Assume now that your method *M* accounts for *30%* of the running time. What Speedup is necessary to cut the overall running time in half?
c) Assume now that the method M can be sped-up by a factor of three. What is the necessary percentage of M to the whole program such that the overall Speedup is doubled?

Part 2

Your application achieves a Speedup S_2 on *two* processors. Use Amdahl's Law to express S_n, the Speedup on *n* processors, in terms of *n* and S_2.

Part 3

You have the choice between buying a single-core CPU or a Multi-Core CPU for a particular application. The single-core CPU is capable of *5* zillion instructions per second, whereas the multi-core CPU has *10* cores, each capable of performing *1* zillion instructions per second. Use Amdahl's Law to derive an explanation for how you would decide, given a particular application that is supposed to be run on your processor.

Part 4

Write a program that is supposed to achieve the maximum number of floating point operations(FLOPS) per second on your particular machine in single-threaded mode. Before you get started, examine your machine to determine the installed CPU and all its characteristics. On linux the command `cat /proc/cpuinfo` can come in handy. Compute the maximum theoretical performance for your CPU when run in single-threaded mode.
Write a program that computes the *axpy function on arrays of size N, for sufficiently large N. Do *k* runs of the same call and measure the time for all of those runs. Vary *k* and *N* in order to determine the maximum number of FLOPS per second you can achieve. Compare your measurement with the value you computed earlier. Explain possible differences.