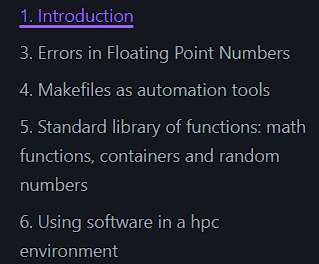
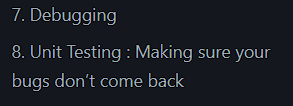
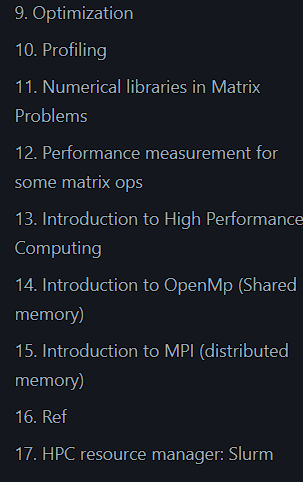
****

****

**Debugging:**

Program debugging is the process of identifying and correcting programming errors.

* **Unit Testing:**Unit testing allows to ensure that a given software behaves in the correct way, at least for the cases one is testing. Once a function is written (or even before in TTD) or a bug is fixed, it is necessary to write a test that ensures the function to work properly in limit cases or the bug to not reappear in the future. There are several levels associated with unit testing .
* **Integral Testing:**

****

**High Performance Computing:**

First, make your program correct, then measure, then optimize. Optimization can be achieved by just using compiler flags, by using already optimized libraries, and sometimes by being careful with our memory usage. Some low level techniques are better left to the compiler.

* **Profiling:**

You need to detect functions which take most of the time. Optimizing a function that takes only 5% of the time will give you only marginal benefits. Finding the functions that take most of the time is called profiling , and there are several tools ready to help you.

* **Benchmarking:**

In this section we will learn how to perform some benchmarks on some matrix operations to get an idea of the operations performance. This will be done manually, but in the future you can use specialized tools like google benchmark, google/benchmark, or nanobench, https://nanobench.ankerl.com/ , etc.

**Good programming practices:**