Mathematicians do two things: computation and reasoning about computation. The aim of computation is to find the answer to a given problem, using a method that works; the aim of reasoning about computation is to find a method that works, and to know that it works, with essentially total reliability. Errors are not tolerable in mathematics since this defeats the purpose of the discipline: mathematics is essentially meant to be a constructive alternative to trial and error.

Computation is the following of an effective procedure to solve a problem. A mathematician, sufficiently drilled so as to follow a given procedure, is one computation device; a laptop such as the one on which I am writing this[1] is another. These two different devices have different strengths and weaknesses, and using one in place of the other naively is going to lead to problems. Modern digital computers are essentially totally reliable electromechanical followers of (numerically coded) instructions. In other words, what a computer really does, so far as it is concerned, is:

> Do thing-numbered-345 with the following numbers: 3, 47 and 2; Do thing-numbered-257 with the following numbers: 3, 47 and 1; Do thing-numbered-345 again with the following numbers: 3, 47 and 3; Do thing-numbered-24 with the result of the last step.

When programmers explicitly write for a digital computer in this way, they use a language called 'assembly language'[2] which makes the generation ofa list of instruction numbers more readable. It looks a little like as follows:

```
; read a byte from stdin
mov eax, 3              ; 3 is recognized by the system as meaning "read"
mov ebx, 0              ; read from standard input
mov ecx, variable       ; address to pass to
mov edx, 1              ; input length (one byte)
int 0x80                ; call the kernel
```

Giving instructions to a human computer will use a different language, and different steps will be available, but it is just as mechanical. To learn to be able to think this way is hard, and in the days before digital computers the most popular method was to threaten schoolchildren with the cane until they learned, by whatever method they could find, a way to rearrange symbols so that they repeatedly got the correct answer. This nightmare has left a lasting memory and a lasting legacy in terms of how people perceive mathematics. Computation is the actual doing of this work, mathematics is really about figuring out the best way of *not* doing this work, given that the problem at hand needs to be solved (so just ignoring it, like a doctor ignoring a dying patient, is just not an option). Mathematics done well saves lives; mathematics done poorly is not worth doing, and may well cost lives. Getting mathematics right is critical

[1] an Acer 5750 if you are interested.

[2] Unless they are really hardcore and hand-calculate the list of instruction numbers.

if it is to be used at all, and the key to getting mathematics right is getting mathematics education right.

Getting mathematics education right should involve looking at available resources and finding the best way to turn learners into mathematicians. Good mathematicians love their subject, and are good at it. Mathematicians who do not love their subject will only do it grudgingly, and will miss those delighfully clever lines of reasoning that massively simplify a problem, and it is those clever lines of reasoning that make mathematics worthwhile. Thus mathematical beauty (that emotional experience mathematicians get when they find an easy way through an apparently hard problem) is not an optional extra: beauty is the ultimate defense against complexity, and complexity is in many ways the enemy of the mathematician, albeit an enemy he (or she) understands well. The beginning stage of mathematical exploration is play, just as the beginning stage of learning for a child is play: the method is identical, and only the scenario changes. Play needs to be fun: fun is that emotional experience which tells you your play appears to be worthwhile. It is the job of the teacher of mathematics to make the initial steps fun, and to keep them fun, until the learner finds enough joy in exploring mathematics that they are self-motivated enough to want to explore it on their own and to apply what they learn to daily life. Computers give a wonderful tool to aid this process.

Once a learner gets to the stage that they are self-motivated enough to want to explore on their own and to apply what they learn, then it is useful to start bringing in real world problems on which to hone these skills, but until a deep inquisitive joy, with both a disciplined mind and playful nature, is present, real world problems are a distraction which will cause the learner to be put off the subject proper. The first stage in playing with a computer is acquiring one. Whilst these days you can buy a laptop that runs Windows, and can reformat it to run Linux if you want a better platform for serious computation, if you want to make a computer out of your own self, and your own body and brain, you need a different approach. I call this approach *the Art of Mental Computation.*

1 The Art of Mental Computation

Mental computation is all about harnessing what human nature gives us to think effectively about problems. Numbers are a useful tool, and equations can be a useful tool too, but they are not the be-all and end-all. What matters is understanding yourself as a human being, understanding how your mental processes work, and learning how to hone them so as to do interesting things. The key lies in those abilities which are normally used for language, and those which are normally used for creative expression. With practice and discipline, through this 'mental computation', mathematics itself becomes another mode of creative self-expression. Then it becomes a joyful delight and it is that joyful delight (as compared to the drudgery normally associated with mathematics) that I really wish to share with you.

When we read words off a page, we are engaging in a process which modern digital comuputers are still struggling to effectively emulate. When we speak, it is likewise. When we listen, carefully, to a stream of words, or read them, and we ponder the meaning, we are doing something so complex that we may never figure out methods by which they may be mechanically emulated. On the other hand, all we are really doing is recognising things we see or hear, and recognising the sensible thing to do with them. Recognition is key, and recognition (re-cognition) is all about knowing what you've processed before.

Much of the reality in which we live works with a mechanical reliability which is almost magical in its nature. It is because of this reliability that we have subjects such as physics, and devices like computers, smartphones and tablets. When things do behave in a mechanically reliable way, they will do the same thing under the same circumstances. Thus knowing what happened last time is of great importance. Our brains are biological machines which are expert at dealing with this: when a brain sees something it has seen before, it reacts to the sameness. When a brain sees something similar to what it has seen before, it reacts to the similarity. Sometimes these reactions have useful outcomes, sometimes not. Almost magically, brains can and do learn in response to these outcomes. But with that comes a potential nightmare of a brain that has learned to think erroneously. Mental computation is also intended as a means to redrill a damaged brain so as to be able to think productively again. It is for this purpose that I developed it for myself, beginning with a brain effectively poisoned by psychiatric 'medication' and with only hints as to how a properly working mathematical mind is supposed to work (that is, memories of my life before 'medication') and a lot of time, effort, anguish, and near-random exploration of various schools of understanding.

The magic of mental computation is that by learning the right recognition processes, you can instinctively and intuitively think, and think playfully, in a way which is logically rigorous. That is, think in a way which makes few errors, and see those errors more often than not. Like a concert pianist, with practice, you can learn a beautiful effortless way of being whose results are a joy to both you and those who are in a position to enjoy the fruits of your efforts.