Diagram 1 serves to point out some imperfections in a majority of chess programs. Let us analyse it attentively (with the Next Best function of Mach III): c8=Q+ promotes a Pawn to a Queen with check but only manages a draw: 1. c8=Q+ Kh7 2. Qc2+ (Kxg2? f3+ Kf1 [if Kxf3 or Kg3 there is a quick mate] Rh4 Qh3) e4 3. Kxg2 (if f3 Qh1+ mate in 7) f3+ 4. Kg1 Qh3 5. Qf1 Qg4+ 6. Kh1 (if Kh2 Be5+) Qh4+ draw.

Qe8+, the only winning move, was found in only 10 seconds with Next Best: 1. Qe8+!! Qxe8 2. Rb8!, winning. All other moves lose quickly.

Let us see what the micros think. The performance of the Mach III is highly typical. It instantly chooses c8=Q+ with a high evaluation; after 9 hours it finds the move leads to a draw +0.00. One might have thought that the computer should find Qe8+ quickly, but it spends twenty seven and a half hours! The reason is that Qe8+ is at the bottom of the move list because it is falsely evaluated to lose. What happens between 9 and 27.5 hours is that, incredibly, Mach III spends its time analyzing Kxg2 and other absurd moves.

Responsibility for this crash lies in the bad sorting of the move list, which pushes Qe8+ to the end of the list when it should be in second place. The quick mate Qh1+ must be at the top of its move list to allow very fast pruning.

Programmers should take heed that with a better sorting of the move list, large amounts of CPU time can be saved in the tree search.

During the 7th WCCC held in Madrid, I was privileged to test Mr. Richard Lang’s program Chess Genius 1.0 with this position. It found the correct move Qe8+ in only 7 minutes and 9 seconds on a 486/66. Assuming a speed up of a factor of 8 in the hardware, this works out at just over 57 minutes on a 286/16, the best time achieved so far.

GUARDS IN HONOUR OF TORRES Y QUEVEDO.
The Universidad Politécnica de Madrid proudly displaying machine-chess’ oldest relic (Madrid, November 1992).