

# MKBN - Mate with King, Bishop and Knight Practice

© 2019 Valentín Albillo

## Abstract

*MKBN is a simple RPN program written in 1980 for the HP-41C programmable calculator and compatibles to help the user practice the basic checkmate with King, Bishop and Knight vs King alone within a given number of moves. Two worked examples included.*

**Keywords:** chess, checkmate, practice, King, Bishop, Knight, KBN vs. K, HP-41C, compatibles, programmable calculator, RPN

## 1. Introduction

*MKBN* is a mid-size (285 steps) RPN program that I wrote in 1980 for the HP-41C programmable calculator and compatibles (will run *as-is* in the HP-41CV and the HP-41CX and with trivial or no changes in other compatible models such as the HP42S, see *Note 1* and also *Note 3*.)

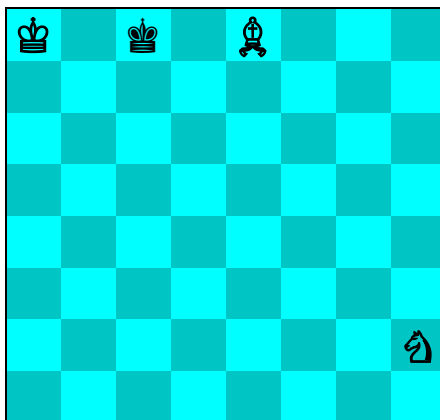
The program is intended to help the user practice in order to achieve the difficult basic checkmate of *King*, *Bishop* and *Knight* (controlled by the user) vs. *King* alone (controlled by the program) within a specified number of moves. The user must try and checkmate before the allotted moves elapse while the program does its best to avoid being checkmated. The possible final outcomes are these:

- the player *checkmates* the enemy King within the specified number of moves. It's a *Win* for the user.
- the player *stalemates* the enemy King within the specified number of moves. The game ends in a *Draw*.
- the enemy king *captures* the Bishop or the Knight. The game ends in a *Draw*.
- the specified number of moves *elapses* before any of the above outcomes. The game ends in a *Draw*.

The program uses a *very fast* (a few seconds per move) and simple but quite effective positional strategy which will frequently succeed in avoiding being checkmated against human players not too experienced with this basic checkmate, and even against *vintage* programs not having access to the appropriate endgame tablebase<sup>1</sup>.

Except in a negligible number of abnormal initial positions, checkmate can always be achieved in *33 moves or less*. As per *FIDE* rules, if this mate appears on the board the winning side has a maximum of **50** moves to try and checkmate the lone king, else the game is considered a *Draw*.

Because of its difficulty and rarity, even very strong human players have at times failed to achieve it in serious competition, much to their embarrassment. The program's strategy isn't perfect (only endgame tablebases are guaranteed to play 100% perfect moves for both sides) so once you've trained long enough to be able to beat it in less than 50 moves for most initial positions, try specifying a lower maximum number of moves, say **35** or even **28** for a much harder challenge, little leeway for errors (see *Examples* below.) This is one of the longest mates:



White to play and checkmate in 33

<sup>1</sup> Of course against an adversary using a KBNK endgame tablebase you can't avoid being checkmated as quickly as possible.

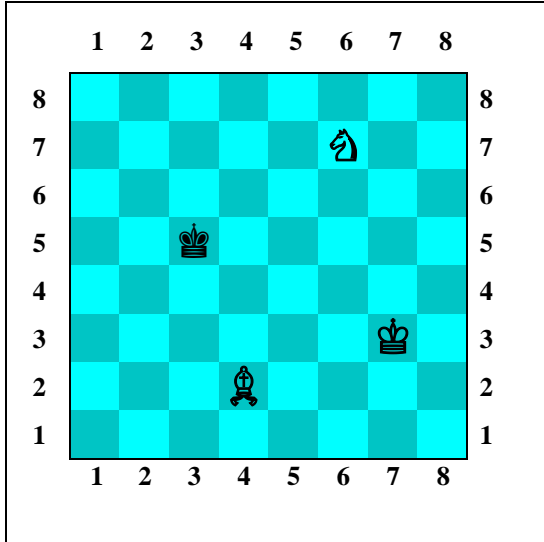
## 2. Program Listing





|    |                   |     |                   |     |               |     |               |     |               |     |                   |
|----|-------------------|-----|-------------------|-----|---------------|-----|---------------|-----|---------------|-----|-------------------|
| 01 | <u>LBL "MKBN"</u> | 51  | 10                | 101 | -1            | 151 | 11            | 201 | RTN           | 251 | "STALE"           |
|    | FIX 0             |     | MOD               |     | GTO 00        |     | X=Y?          |     | RCL 09        |     | FS? 00            |
|    | CF 29             |     | X>=Y?             |     | <u>LBL 05</u> |     | RTN           |     | X#0?          |     | "CHECK"           |
|    | CLX               |     | ISG Z             |     | -10           |     | RCL 07        |     | SIGN          |     | ┌"MATED IN "      |
| 05 | "MY KING IN"      | 55  | <u>LBL 00</u>     | 105 | GTO 00        | 155 | RCL 03        | 205 | 11            | 255 | ARCL 04           |
|    | PROMPT            |     | XEQ IND Z         |     | <u>LBL 06</u> |     | -             |     | <u>LBL 09</u> |     | ┌" MOVES"         |
|    | STO 00            |     | FS? 00            |     | -9            |     | ABS           |     | *             |     | <u>LBL 11</u>     |
|    | "YOUR KING IN"    |     | GTO 19            |     | GTO 00        |     | 8             |     | STO 10        |     | AVIEW             |
|    | PROMPT            |     | RCL 07            |     | <u>LBL 07</u> |     | X=Y?          |     | RCL 07        |     | BEEP              |
| 10 | STO 01            | 60  | STO 00            | 110 | 9             | 160 | RTN           | 210 | RCL 10        | 260 | RTN               |
|    | "BISHOP IN"       |     | ">"               |     | GTO 00        |     | X<>Y          |     | RCL 02        |     | <u>LBL 14</u>     |
|    | PROMPT            |     | ARCL 04           |     | <u>LBL 08</u> |     | 12            |     | +             |     | 84567231          |
|    | STO 02            |     | ┌": I PLAY "      |     | -11           |     | X=Y?          |     | X=Y?          |     | GTO 00            |
|    | "KNIGHT IN"       |     | ARCL X            |     | <u>LBL 00</u> |     | RTN           |     | RTN           |     | <u>LBL 15</u>     |
| 15 | PROMPT            | 65  | TONE 9            | 115 | RCL 00        | 165 | X<>Y          | 215 | STO 08        | 265 | 65218347          |
|    | STO 03            |     | PROMPT            |     | +             |     | 19            |     | RCL 07        |     | GTO 00            |
|    | "MAX. MOVES ?"    |     | "BISHOP "         |     | STO 07        |     | X=Y?          |     | RCL 10        |     | <u>LBL 16</u>     |
|    | PROMPT            |     | RCL 02            |     | <u>LBL 13</u> |     | RTN           |     | -             |     | 73481526          |
|    | STO 04            |     | RCL 07            |     | SF 00         |     | X<>Y          |     | STO 06        |     | GTO 00            |
| 20 | 1 E3              | 70  | X=Y?              | 120 | 11            | 170 | 21            | 220 | CF 01         | 270 | <u>LBL 17</u>     |
|    | ST/ 04            |     | GTO 09            |     | RCL 07        |     | X=Y?          |     | RCL 10        |     | 12376458          |
|    | ISG 04            |     | "KNIGHT "         |     | X<Y?          |     | RTN           |     | X>0?          |     | <u>LBL 00</u>     |
|    | <u>LBL 12</u>     |     | RCL 03            |     | RTN           |     | RCL 07        |     | SF 01         |     | STO 05            |
|    | "YOUR MOVE..."    |     | X=Y?              |     | 88            |     | RCL 02        |     | <u>LBL 10</u> |     | <u>LBL 18</u>     |
| 25 | PROMPT            | 75  | GTO 09            | 125 | X<Y?          | 175 | X=Y?          | 225 | RCL 03        | 275 | RCL 05            |
|    | <u>LBL A</u>      |     | "ILLEGAL KING"    |     | RTN           |     | CF 00         |     | RCL 08        |     | INT               |
|    | "KING TO ?"       |     | ┌" POSITION"      |     | RCL 07        |     | X=Y?          |     | CF 00         |     | X=0?              |
|    | PROMPT            |     | X<>Y              |     | 10            |     | RTN           |     | X=Y?          |     | RTN               |
|    | STO 01            |     | RCL 01            |     | MOD           |     | -             |     | RTN           |     | 10                |
| 30 | GTO 00            | 80  | X=Y?              | 130 | X=0?          | 180 | STO 09        | 230 | RCL 01        | 280 | ST/ 05            |
|    | <u>LBL B</u>      |     | GTO 11            |     | RTN           |     | 11            |     | X=Y?          |     | MOD               |
|    | "BISHOP TO ?"     |     | CIA               |     | 9             |     | /             |     | RTN           |     | XEQ IND X         |
|    | PROMPT            |     | ARCL 04           |     | X=Y?          |     | STO 10        |     | SF 00         |     | FS? 00            |
|    | STO 02            |     | ┌" MOVES: DRAW"   |     | RTN           |     | 9             |     | RCL 10        |     | GTO 18            |
| 35 | GTO 00            | 85  | ISG 04            | 135 | RCL 07        | 185 | ST/ 09        | 235 | ST+ 08        | 285 | <b>END</b>        |
|    | <u>LBL C</u>      |     | GTO 12            |     | RCL 01        |     | RCL 09        |     | RCL 06        |     |                   |
|    | "KNIGHT TO ?"     |     | GTO 11            |     | -             |     | FRC           |     | RCL 08        |     |                   |
|    | PROMPT            |     | <u>LBL 09</u>     |     | ABS           |     | X#0?          |     | FC? 01        |     | <b>Uses:</b>      |
|    | STO 03            |     | ┌"CAPTURED: DRAW" |     | 1             |     | GTO 00        |     | GTO 00        |     | - 285 steps       |
| 40 | <u>LBL 00</u>     | 90  | GTO 11            | 140 | X=Y?          | 190 | RCL 09        | 240 | X<=Y?         |     | - sets FIX 0      |
|    | RCL 00            |     | <u>LBL 01</u>     |     | RTN           |     | X#0?          |     | GTO 10        |     | - flags 0-1, 29   |
|    | 50                |     | 11                |     | X<>Y          |     | SIGN          |     | RTN           |     | - labels 00-19    |
|    | X>Y?              |     | GTO 00            |     | 9             |     | 9             |     | <u>LBL 00</u> |     | and A-C           |
|    | 0                 |     | <u>LBL 02</u>     |     | X=Y?          |     | GTO 09        |     | X>=Y?         |     | - registers 00-10 |
| 45 | 25                | 95  | 1                 | 145 | RTN           | 195 | <u>LBL 00</u> | 245 | GTO 10        |     |                   |
|    | /                 |     | GTO 00            |     | X<>Y          |     | RCL 10        |     | RTN           |     | - requires at     |
|    | 14                |     | <u>LBL 03</u>     |     | 10            |     | FRC           |     | <u>LBL 19</u> |     | least 1 Memory    |
|    | +                 |     | 10                |     | X=Y?          |     | X#0?          |     | RCL 00        |     | Module to run     |
|    | 5                 |     | GTO 00            |     | RTN           |     | CF 00         |     | STO 07        |     | in the original   |
| 50 | RCL 00            | 100 | <u>LBL 04</u>     | 150 | X<>Y          | 200 | X#0?          | 250 | XEQ 13        |     | HP-41C.           |

Note: to enter text lines use **ALPHA**, ┌ is the **Append** character; / is the **Divide** key and \* is the **Multiply** key.

### 3. Usage Instructions





The program uses a *column/row* convention to specify the coordinates of the square where a piece is currently located or the square where it moves to, where *column* and *row* are single-digit<sup>1</sup> from 1 to 8. E.g.: in the figure:



- the White King  is located at **73** (col. 7, row 3)
- the White Bishop  is located at **42** (col. 4, row 2)
- the White Knight  is located at **67** (col. 6, row 7)
- the Black King  is located at **35** (col. 3, row 5)

To run a practice session, proceed as follows:

Step 1: Input the initial position and max. # of moves to mate:

|                                                                                              |                                  |   |              |
|----------------------------------------------------------------------------------------------|----------------------------------|---|--------------|
| <input type="text" value="XEQ"/>                                                             | "MKBN"                           | → | MY KING IN   |
| col/row of  | <input type="text" value="R/S"/> | → | YOUR KING IN |
| col/row of  | <input type="text" value="R/S"/> | → | BISHOP IN    |
| col/row of  | <input type="text" value="R/S"/> | → | KNIGHT IN    |
| col/row of  | <input type="text" value="R/S"/> | → | MAX. MOVES ? |
| Max. # of moves                                                                              | <input type="text" value="R/S"/> | → | YOUR MOVE... |

Step 2: In **USER** mode, execute the appropriate subroutine  (King),  (Bishop) or  (Knight) to specify the piece you want to move and the coordinates *col/row* of the square where you're moving it to:

**Important:** If the initial position is legal, and the user always plays legal moves, the program will *never* play an illegal move either. However, *it doesn't check either your moves or the initial position for legality* so you should make absolutely sure your moves are fully legal before entering them (check for wrong piece or wrong location to move it to) or the current game might get *corrupted* and would have to be amended. See **Note 2**.

→ KING TO ? **or**  → BISHOP TO ? **or**  → KNIGHT TO ?

col/row  → >I: I PLAY (col/row) { take note of the program's reply and press  to continue }

→ either one of the six possible **final results** below **or** YOUR MOVE...

Step 3: Repeat Step 2 above to enter your moves and get the program's replies, until one of these *final* results:

|                               |                                                                                    |
|-------------------------------|------------------------------------------------------------------------------------|
| <b>CHECKMATED IN nn MOVES</b> | You succeeded in giving checkmate within the specified moves. A <b>Win</b> .       |
| <b>nn MOVES: DRAW</b>         | You failed in giving checkmate within the specified moves. A <b>Draw</b> .         |
| <b>STALEMATED IN nn MOVES</b> | You stalemated the enemy King. A <b>Draw</b> .                                     |
| <b>BISHOP CAPTURED: DRAW</b>  | The enemy King captured your Bishop. A <b>Draw</b> .                               |
| <b>KNIGHT CAPTURED: DRAW</b>  | The enemy King captured your Knight. A <b>Draw</b> .                               |
| <b>ILLEGAL KING POSITION</b>  | You moved your King next to the enemy King so it was "captured". <b>Null</b> game. |

Step 4: To practice once more giving checkmate in this same position or a different one, go to Step 1 above.

<sup>1</sup> The standard algebraic notation for chess, where columns are specified by a single *letter* from *a* to *h* instead of a single *digit*, isn't used in this program because the version of *RPN* implemented in the *HP-41C/CV* doesn't include *any* string-handling functions other than storage/recall/append, comparison for equality and simple input/output. In particular there's no simple way to extract characters from a string or convert a character to its ASCII equivalent so processing inputs such as *e4* or *b7* (let alone *Kb7* or *Bh3*), while doable, would be cumbersome and wasteful of memory resources.

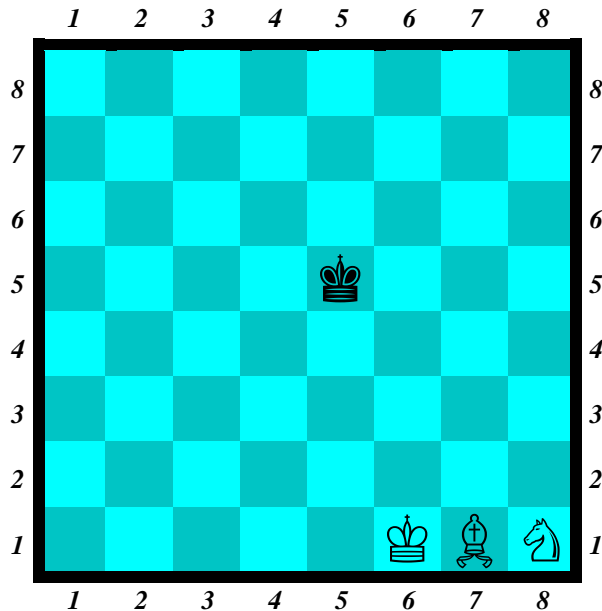
## 4. Examples

The following examples can be useful to check that the program is correctly entered and to understand its usage.

### 4.1 Example 1

Playing the White pieces and moving first, try and checkmate the lone Black King starting from this position:

*FEN 8/8/8/4k3/8/8/8/5KBN/ w*



To conduct the winning forces against our *RPN* practice program running on a battery-operated programmable calculator released in 1980, the *HP-41C* (proprietary *CMOS CPU @ 355 KHz*), we'll use the powerful *Stockfish 9 (2018)* chess engine (>3,300 ELO, far exceeding Grandmaster level<sup>1</sup>) running at about a million nodes per second on a mid-range *Samsung* tablet, with a hash table of 1 Gbyte (but no endgame tablebases, that would be unfair) and evaluating slightly over 100 million nodes before committing its move.

The given sample position shown left is actually a *Mate in 30* with perfect play by both sides according to the *KBNK Nalimov* tablebase, so let's specify **30** moves as the maximum limit and see how both contenders fare ...

In **USER** Mode, proceed as follows:

```

XEQ "MKBN" → MY KING IN { remember, positions and moves are entered as column/row }
55 R/S → YOUR KING IN
61 R/S → BISHOP IN
71 R/S → KNIGHT IN
81 R/S → MAX. MOVES ?
30 R/S → YOUR MOVE...
A → KING TO ?
52 R/S → >1: I PLAY 45 { take note of the reply and always press R/S to continue }
R/S → YOUR MOVE...
A → KING TO ?
43 R/S → >2: I PLAY 55 { Stockfish 9 evaluates its move as giving mate in 36 (or less) }
R/S → YOUR MOVE...
A → KING TO ?
34 R/S → >3: I PLAY 54 { Stockfish 9 evaluates its move as giving mate in 33 (or less) }

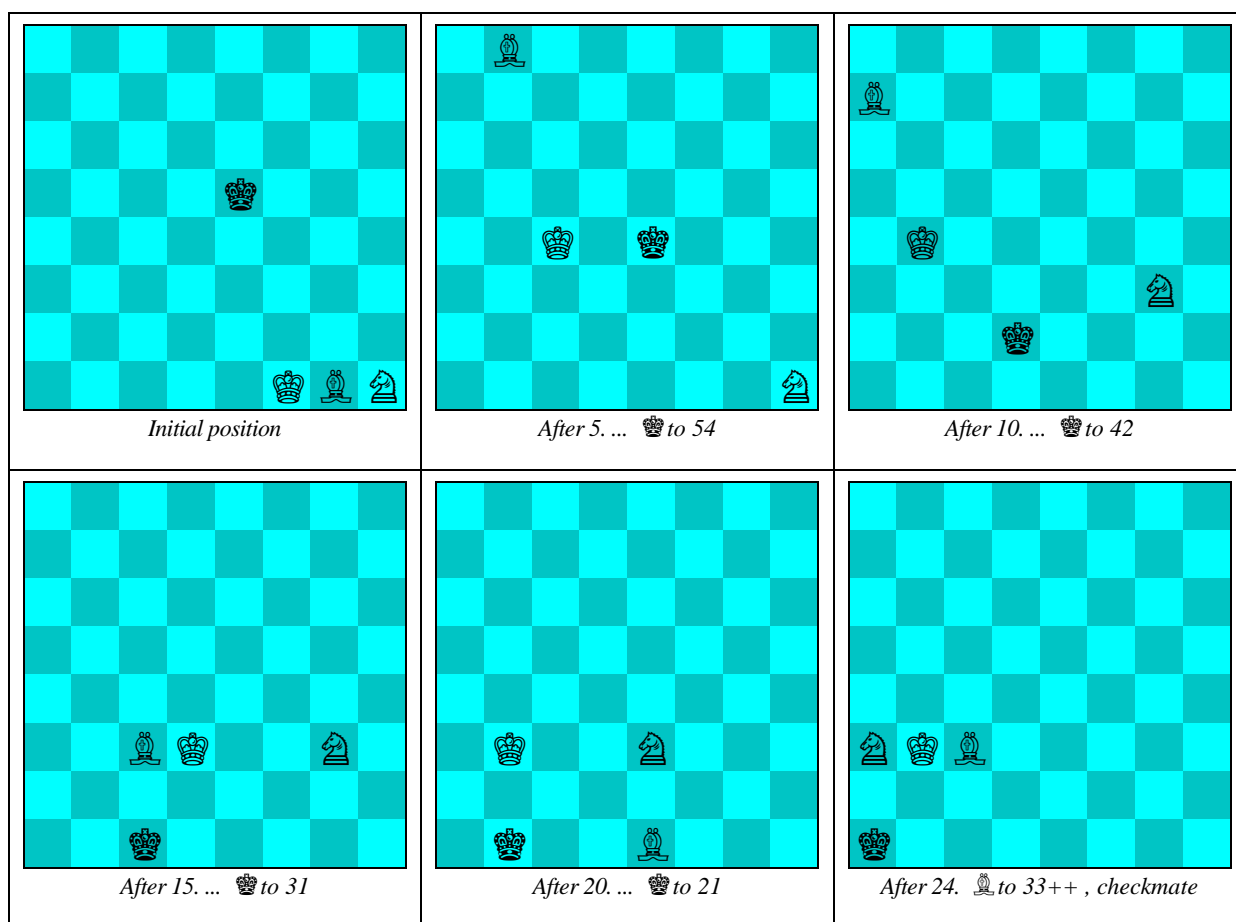
```

<sup>1</sup> At the time of writing (2019) all human Grandmasters are rated at less than 3,000 ELO (e.g., Magnus Carlsen: 2,882 ELO)

The game goes on with the moves and *SF* evaluations given in the table (*exact Nalimov KBNK eval. in parentheses*):

| #  | Stockfish 9 | Eval. by SF9            | MKBN      | #  | Stockfish 9 | Eval. by SF9 | MKBN                      |
|----|-------------|-------------------------|-----------|----|-------------|--------------|---------------------------|
| 4  | B to 17     | Mate in 31 (28)         | I PLAY 55 | 15 | B to 33     | Mate in 10   | I PLAY 31                 |
| 5  | B to 28     | Mate in 29 (27)         | I PLAY 54 | 16 | N to 65     | Mate in 9    | I PLAY 41                 |
| 6  | N to 73     | Mate in 27 (26)         | I PLAY 53 | 17 | N to 53     | Mate in 8    | I PLAY 31                 |
| 7  | K to 45     | Mate in 26 (25)         | I PLAY 43 | 18 | K to 34     | Mate in 7    | I PLAY 21                 |
| 8  | B to 17     | Mate in 17 <sup>1</sup> | I PLAY 33 | 19 | K to 23     | Mate in 6    | I PLAY 31                 |
| 9  | K to 35     | Mate in 16              | I PLAY 43 | 20 | B to 51     | Mate in 5    | I PLAY 21                 |
| 10 | K to 24     | Mate in 15              | I PLAY 42 | 21 | B to 42     | Mate in 4    | I PLAY 11                 |
| 11 | K to 34     | Mate in 14              | I PLAY 51 | 22 | N to 32     | Mate in 3    | I PLAY 21                 |
| 12 | B to 53     | Mate in 13              | I PLAY 41 | 23 | N to 13     | Mate in 2    | I PLAY 11                 |
| 13 | K to 43     | Mate in 12              | I PLAY 51 | 24 | B to 33     | Mate in 1    | CHECKMATED<br>IN 24 MOVES |
| 14 | B to 44     | Mate in 11              | I PLAY 41 |    |             |              |                           |

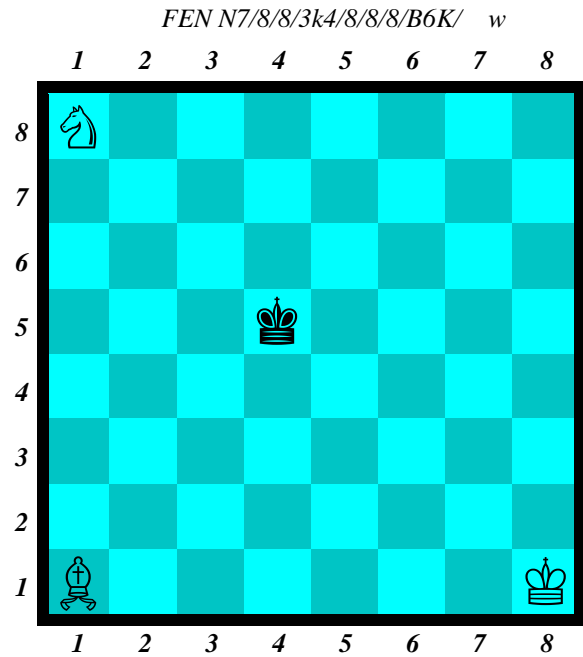
Thus, *Stockfish 9* has *checkmated* the black King in just **24** moves, less than the specified 30 move limit, but *MKBN* resisted bravely and indeed its last 16 moves were *optimal*. Some positions from the game:



<sup>1</sup> The previous *move #7* by Black was far from optimal, shortening White's future checkmate by 8 moves, but afterwards *all* 16 subsequent Black's moves are 100% perfect as per *Nalimov KBNK* and delay the checkmate as much as possible.

#### 4.2 Example 2

Playing the White pieces and moving first, try and checkmate the lone Black King starting from this position:



This time we won't use a powerful program to deliver the mate but we'll let the human user try instead.

The given sample position shown left is also a *Mate in 30* with perfect play by both sides, so we'll specify **35** moves as the maximum limit for the user to try and checkmate ... or not !

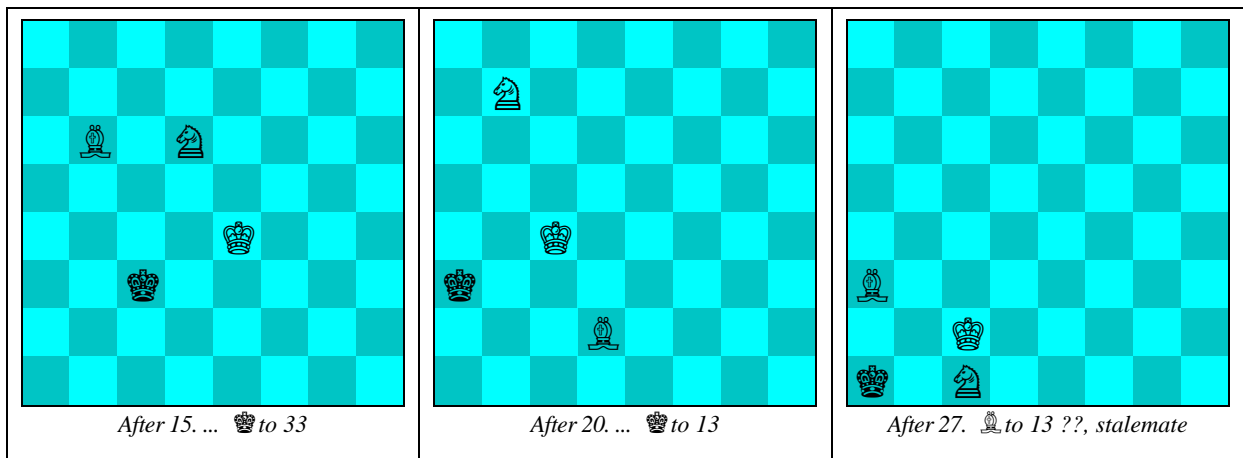
In **USER** Mode, proceed as follows:

|     |          |                       |
|-----|----------|-----------------------|
| XEQ | "MKBN" → | <i>MY KING IN</i>     |
| 45  | R/S      | → <i>YOUR KING IN</i> |
| 81  | R/S      | → <i>BISHOP IN</i>    |
| 11  | R/S      | → <i>KNIGHT IN</i>    |
| 18  | R/S      | → <i>MAX. MOVES ?</i> |
| 35  | R/S      | → <i>YOUR MOVE...</i> |

The game goes on with the moves given in the table below:

| # | User           | MKBN             | #  | User           | MKBN             | #  | User           | MKBN             | #  | User           | MKBN               |
|---|----------------|------------------|----|----------------|------------------|----|----------------|------------------|----|----------------|--------------------|
| 1 | <i>K to 72</i> | <i>I PLAY 54</i> | 8  | <i>N to 43</i> | <i>I PLAY 32</i> | 15 | <i>B to 26</i> | <i>I PLAY 33</i> | 22 | <i>K to 43</i> | <i>I PLAY 13</i>   |
| 2 | <i>N to 26</i> | <i>I PLAY 43</i> | 9  | <i>N to 22</i> | <i>I PLAY 23</i> | 16 | <i>B to 15</i> | <i>I PLAY 23</i> | 23 | <i>K to 32</i> | <i>I PLAY 12</i>   |
| 3 | <i>K to 63</i> | <i>I PLAY 42</i> | 10 | <i>K to 43</i> | <i>I PLAY 24</i> | 17 | <i>K to 43</i> | <i>I PLAY 14</i> | 24 | <i>B to 24</i> | <i>I PLAY 11</i>   |
| 4 | <i>N to 34</i> | <i>I PLAY 43</i> | 11 | <i>N to 34</i> | <i>I PLAY 25</i> | 18 | <i>B to 42</i> | <i>I PLAY 23</i> | 25 | <i>N to 23</i> | <i>I PLAY 12</i>   |
| 5 | <i>N to 55</i> | <i>I PLAY 42</i> | 12 | <i>B to 62</i> | <i>I PLAY 24</i> | 19 | <i>N to 27</i> | <i>I PLAY 14</i> | 26 | <i>N to 31</i> | <i>I PLAY 11</i>   |
| 6 | <i>B to 44</i> | <i>I PLAY 51</i> | 13 | <i>N to 46</i> | <i>I PLAY 15</i> | 20 | <i>K to 34</i> | <i>I PLAY 13</i> | 27 | <i>B to 13</i> | <i>STALEMATED</i>  |
| 7 | <i>K to 53</i> | <i>I PLAY 41</i> | 14 | <i>K to 54</i> | <i>I PLAY 24</i> | 21 | <i>N to 35</i> | <i>I PLAY 22</i> |    |                | <i>IN 27 MOVES</i> |

The user's final move was a terrible blunder, *stalemating* the Black King, a **Draw**. The correct winning move was *B to 33* to which Black's reply would be *CHECKMATED IN 27 MOVES*. Some positions from the game:



## Notes

1. The program can be easily adapted to run under other *RPN* versions with minimal or no changes. For example, the only changes required for it to run on the *HP42S* is simply to change the names of some of the instructions such as **FIX 0** to **FIX 00**, **\*** to **x**, **/** to **÷**, **ST+ / ST- / ST\* / ST/** to **STO+ / STO- / STOx / STO÷**, respectively, **INT** to **IP**, **FRC** to **FP**, **ISG Z** to **ISG ST Z** and so on. See the *HP42S Owner's Manual*, in particular the section titled "Using *HP-41C Programs*" for full details. The resulting program size in the *HP42S* should be 643 bytes.
2. As stated in the *caveat* at page 3, if the initial position is legal and the user always plays legal moves, the program will *never* play an illegal move either, but actually *it doesn't check your moves for legality*. If the program happens to play an illegal move, then (program bugs aside for the time being) it's surely the case that either the position is currently illegal, or you played an illegal move, or both. You can check the current locations of all pieces like this:

With the program halted, execute

|                |   |                                                         |
|----------------|---|---------------------------------------------------------|
| <b>VIEW 00</b> | → | shows the current location of the program's <i>King</i> |
| <b>VIEW 01</b> | → | shows the current location of the user's <i>King</i>    |
| <b>VIEW 02</b> | → | shows the current location of the user's <i>Bishop</i>  |
| <b>VIEW 03</b> | → | shows the current location of the user's <i>Knight</i>  |

If any of these locations happen to be wrong, simply store the correct locations into the respective storage registers (say for instance, **76** **STO 01** ), and continue executing the program by issuing the command **XEQ 12**. This will immediately ask for your move and hopefully the game will then proceed correctly. Else, you'll just have to restart the game anew.

3. I also created at the time an equivalent *BASIC* version of this program for the *SHARP PC-1211* pocket computer which will also run *as-is* in compatible models such as the *SHARP PC-1212* and with minimal changes in the *SHARP PC-1350/1360* and many others. See *References*.

## References

|                                  |                                                                             |
|----------------------------------|-----------------------------------------------------------------------------|
| John Nunn (2009)                 | <i>Understanding Chess Endgames</i>                                         |
| Karsten Müller (2001)            | <i>Fundamental Chess Endings</i>                                            |
| Yuri Averbakh (1966)             | <i>Chess Endings: Essential Knowledge</i>                                   |
| Yuri Averbach (1979)             | <i>Finales de alfil y de caballo (Spanish)</i>                              |
| Ronald Cohn <i>et al.</i> (2012) | <i>Stockfish</i>                                                            |
| Valentin Albillo (1980)          | <i>MATEKBN - Mate with King, Bishop and Knight Practice (SHARP PC-1211)</i> |

## Copyrights

Copyright for this paper and its contents is retained by the author. Permission to use it for non-profit purposes is granted as long as the contents aren't modified in any way and the copyright is acknowledged.