BREAKTHROUGH!

Possible Additional Programming Tasks

1. Create an extra toolkit (e.g. 'd') and add locks involving this to the locks.txt file.

2. Introduce a *Swiss Army Knife* card which can be used as any single card, i.e. any tool from any toolkit.

3. Add the functionality to allow a player, at the start of a turn, to (U)ndo their last turn. This should revert everything – including the score, sequence, hand and discard pile – to the state it was at the start of the previous turn. There should be one undo available per lock and it should not be possible to use it on the first turn of a new lock.

4. Add a *High Scores* file and ability to view this from a main menu.

5. Add levels so that different locks have different challenges which will be more or less difficult depending on the current level. This could be linked to (11 – complexity) or it could be the number of toolkits used, e.g. 2, 3 or 4.

6. Add a *Mighty Hammer* card that can smash (solve) the current lock or you can keep the card in your hand and play it later.

7. Introduce a user-defined locks option. This generates a rough pseudo two-player game where one player can choose a lock sequence and another has to try to unpick it (similar in a way to the original game of *Mastermind*). A user-defined lock must follow the criteria of 3–7 cards, of which at least two must be files, and at least one must be a pick.

8. Introduce a second type of lock – a 'Maths Lock', whereby the player can get some locks as per the way they are now and some new maths locks. This will involve introducing a new attribute for each card called 'Number'. The value of 'number' is displayed in addition to the type of card (pick, file, etc.) currently displayed. Cards can be used for their mathematical attribute rather than their type attribute. For example, if a lock contains four files – each with a number value of 5 – that gives a total lock value of 20. The player needs to play a sequence of cards to also generate a total value of 20. For example, if the player plays two picks, each with a number value of 10, then the lock will open. These new 'Maths Locks' are solved only using the numeric value and are independent of the tool type and tool kit.

9. Receive a bonus of 50 if you quit and the current challenge could never be solved (with the cards and deck as they are currently).

10. Add an *Autoplay* mode which shows a computer simulation of the game.

11. Design a formula to compute a complexity value for a lock.

12. Validation of card to play (with exception handling) for choosing which card to throw away in response to a difficulty card.

13. Validation on entry of choice (or any entry) so that the player can only choose 'D' or 'U' each turn.

14. Be able to sacrifice a card (removed from the game) in order to change the toolkit of any card.

15. Examine the game1.txt file (or game) closely and draw a flow diagram of the possible choices the player could make to move from the 'saved state' to 'end of the game'. Complete this purely from looking at the data in the game1.txt file rather than playing the game.