Implementing and Solving Rubik's Family Cubes with Marked Centres

Ken Fraser[1]

Date of original document: 11 May 2012
Date of this revision: 12 February 2017

Summary

Most implementations of Rubik's family cubes have no distinguishing marks on centre cube elements and these elements can occupy various positions for the solved state. The addition of distinguishing marks to centre cube elements results in there being only one correct positional arrangement of centre cube elements to satisfy the solved state for the cube. The marking adds additional complexity to cube solving and additional algorithms need to be defined.

The author has developed a marked centres' software implementation referred to as "Unravel" that is applicable for cubes in the size range 3x3x3 to 32x32x32 typically if a numerical graphic in the range '1' to '4' is used. This latest version includes a simple corner marking extension that can extend the range to the program limit of 99x99x99 but the author would be surprised if anybody would have an inclination to wish to solve cubes of such large size.

While there are specific references herein to the author's "Unravel" implementation, the design techniques and special algorithms presented are applicable to any other hardware or software implementation.

---

[1] Ken Fraser retired in 2002 as Principal Research Scientist and head of Helicopter Life Assessment at the Aeronautical and Maritime Research Laboratory (as it was known at the time), Defence Science and Technology Organisation, Department of Defence, Australia. This publication is the result of a leisure activity and has no relation to work at the Laboratory.

Contents

1. Definitions

Cube size	The standard Rubik's cube is often referred to as a 3x3x3 cube. That cube will be referred to as a size 3 cube and in general an *n* x *n* x *n* cube will be referred to as a size *n* cube.
Rubik cube family	Cubes that have similar rotational properties to the standard Rubik's size 3 cube and obey generalized rules for a size *n* cube are considered to be members of the Rubik cube family. Cubes of size 2 and above are available in unit size steps in the *Unravel* program. The program limit is size 99 but such a large size or large sizes in general may be too difficult for many users to handle.
Hardware cube	A hardware or physical cube is a Rubik's family cube that comes as a single-size hand-held object.
Software cube	A software cube is a program that emulates and presents the cube in some form on a computer monitor and allows the user to rearrange it. Software cubes that accommodate a range of cube sizes are available. Such cubes are not subject to the physical restraints that impose a significant size limit on hardware forms.
Rule	One of a set of generalized laws that defines what is and what is not possible (usually in mathematical terms) for Rubik's family cubes.
Cubie	Individual cube elements will be referred to as cubies (others sometimes refer to them as "cubelets"). There are three types of cubies: corner cubies (three coloured surfaces), edge cubies (two coloured surfaces) and centre cubies (one coloured surface). The absolute centre cubies for odd size cubes sit on the central axes of the six faces and their relative positions never change.
Cubicle	A cubicle is the compartment in which a cubie resides. For a permutation (defined below), cubicles are considered to occupy fixed positions relative to the cube object but their contents (cubies) may change.
Facelet	A facelet is a visible coloured surface of a cubie (corner cubies have three facelets, edge cubies have two and centre cubies have one).
Cube state	A particular arrangement of the cubies will be referred to as a cube state. What looks the same is considered to be the same (unless specific mention to the contrary is made). Each state has equal probability of being produced after a genuine random scrambling sequence. A rotation of the whole cube does not change the state considered herein. In other texts the various states are often referred to as permutations or arrangements.
Cube layer	A cube layer is a one cubie width slice of the cube perpendicular to its axis of rotation. Outer layers (faces) contain more cubies than inner layers. For a cube of size *n* there will be *n* layers along any given axis.
Cube face	The meaning of a cube face depends on the context in which it is used. It usually means one of the six three-dimensional outer layers but can also refer to just the outside layer's surface which is perpendicular to its axis of rotation. The faces are usually designated as up (U), down (D), front (F), back (B), left (L) and right (R).
Cube style	Two cube styles are referred to in this document: firstly a standard cube with unmarked centres and secondly a cube with marked centres.

Set state	The set (or solved) state of the cube is one for which a uniform colour appears on each of the six faces. For cubes with marked centres the set state also includes a unique arrangement of all centre cubies.
Scrambled state	The scrambled state is the starting point for unscrambling the cube. It arises when a cube in the set or any other state is subject to a large number of randomly chosen layer rotations.
"Fixed-in-space" axes of rotation	There are three mutually perpendicular axes of rotation for the cube. One set of axes defined in terms of D, U, B, F, L and R terms can be considered to have a fixed orientation in space. Think of these axes as belonging to a cube-shaped container in which the cube object can be positioned in any of 24 orientations. One axis can be drawn through the centres of the D and U faces (the DU axis). The others are the BF and LR axes. The main display of the cube for the *Unravel* program uses these axes.
"Cube object" axes of rotation	Another set of axes, can be defined for the cube object itself. These axes relate to the face colours, which are off-white, red, light-orange, green, light-blue and violet for the Unravel's default colour set. The axes are white-blue, red-violet and orange-green. For odd size cubes these axes are always fixed relative to the internal frame of the cube object via the absolute centre cubies. For even size cubes these axes remain fixed relative to the internal frame of the cube object after initial selections. The origin for the axes is the centre of the cube object. It is likely that most users would make these axes coincide with the previous fixed-in-space set as the final solution is approached.
Layer rotation	The only way that cube state can be changed is by rotations of cube layers about their axes of rotation. All changes of state involve rotation steps that can be considered as a sequence of single layer quarter turns.
Orbit	For a basic quarter turn of a cube layer for cubes of all sizes, sets-of-four cubies move in separate four-cubicle trajectories. When all the possible trajectories for a given cubie type are considered for the whole cube we will refer to all the possible movement positions as being in a given orbit. We consider that the size 3 cube has two orbits, one in which the eight corner cubies are constrained to move and one in which the 12 edge cubies are constrained to move. Transfer of cubies between these orbits is not possible. For cubes of size 4 and above we will also define an edge cubie orbit as comprising 12 cubies but will use the term *complementary orbit* to describe a pair of orbits between which edge cubies can move. A pair of complementary edge cubie orbits contains a total of 24 cubies. Cubes of size 4 and above include centre cubie orbits that contain 24 cubies. Transfer of cubies between one such orbit and another is not possible (applies to cubes of size 5 and above).
Move	A move is a quarter turn rotation of a layer or a sequence of such quarter turns that a person would apply as a single step.
Move notation	A clockwise quarter turn of an outer layer is usually expressed as U, D, F, B, L or R. In other respects the notation used varies among authors. The move notation used in this document is defined in Appendix A.
Algorithm	An algorithm defines a sequence of layer rotations to transform a given state to another (usually less scrambled) state. Usually an algorithm is expressed as a printable character sequence according to some move notation. An algorithm can be considered to be a “smart” move. All algorithms are moves but few moves are considered to be algorithms.

Permutation	A permutation of the cube as used herein means the act of permuting (i.e. rearranging) the positions of cubies. A permutation is an all-inclusive term which includes all moves (and algorithms) and more. Even the solving of the cube from a scrambled state represents a permutation. The term "permutation" is used extensively by mathematicians who use Group Theory to quantify the process involved in a rearrangement of cubies. The term "permutation" is also often used to mean the state of the cube that results after it is rearranged but that meaning will not be used herein. In such cases the term "cube state" will be used. That allows the term "permutation" to be used when the permutation results in no change of state – an area of special interest for Rubik's family cube permutations.
Parity	A cube permutation can be represented by a number of swaps of two cubies. If that number is even the permutation has even parity, and if the number is odd the permutation has odd parity.

2. Introduction

The focus for this document is the description of a software means of implementing Rubik's family cubes with centre cubire markings over a large size range and indicating a means of solving them using that or any other implementation in either hardware or software form. Except to a minor extent, cube mathematics is not included in this document although the author's original intention for investigating cubes with marked centres was to verify the mathematics presented in other documents. Cube mathematics, including that relevant to cubes with marked centres, is provided in two of the author's documents listed in the References section. Those documents have contents as summarised in the following table and provide references to relevant works by other authors.

Reference	Content matter
First document[1]	The number of reachable states for standard cubes with unmarked centres and for cubes with marked centres.
Second document[2]	Parity and orientation rules for cubes with unmarked and marked centres. Permutation cycle length. Relationship between reachable and unreachable states.

The contentious mathematical issue that prompted the author to seek a practical confirmation was the number of reachable states factor for a 24 cubie orbit for cubes of size 4 and above with marked centres. The commonly stated figure by other authors is `24!` (24 factorial)[2]. The author considers this figure does not comply with parity or orientation rules and should be `24!`/2 as used in the author's mathematical documents[1][2]. To verify the proposition in a practical sense all that is required is to demonstrate that when all centre cubies in a given orbit except the last four are in place, that there are only 12 possibilities (`4!`/2) that can be reached without changing anything except the positions of these last four centre cubies. The contrary view would postulate that there are 24 possibilities (`4!`).

Using the author's *Unravel* (Java version) software implementation[3], it was confirmed that for the theoretical 24 possible arrangements for the last four centre cubies in a giver 24 centre cubie orbit, 12 states are reachable and 12 are unreachable. To enable this demonstration, a

2 In mathematics, the factorial of a positive integer *n*, denoted by *n*!, is the product of all positive integers less than or equal to *n*. For example, `4! = 4x3x2x1` (equals 24).

method of marking centre cubies needed to be developed and a method of solving cubes with marked centres had to be accomplished. At the time of writing the author was unable to find anybody else who had completed a similar development.

Although the mathematical issue referred to above could be resolved with just a size 4 cube with marked centres, the scope of the development was extended to cubes of larger size.

3. Method for Implementing Marked Centres in Rubik's Family Cubes

3.1 Design Objectives

For any implementation the following objectives must be met:

- For the solved cube, all cubies must end up in exactly the same relative positions (including orientations for absolute centre cubie for odd size cubes) as for the original set state.
- Full compliance with cube mathematical rules must be achieved.

Typically, hardware cubes with marked centres use images or logos on the faces to designate what centre cubie(s) orientation is required for a solved cube. Application of such markings is generally restricted to just cubes of very small size. Some implementations of marked size 3 hardware cubes don't have markings on all centre cubies. Partial implementations are not considered in this document or in the *Unravel* program.

For the specific *Unravel* program implementation the following special requirements are added:

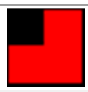
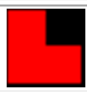
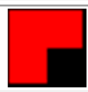
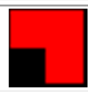
- The centre cubie marking must be as simple as possible.
- The program must accommodate both unmarked and marked centres with the choice readily selectable at run-time.
- Since the *Unravel* program tracks elapsed time from a scrambled start, it must stop the clock when the solved state is detected. Unlike the standard cubes with unmarked centres for which the solved state is signified by a uniform color on all six faces, the marked centres'case requires more involved checking of all 24 spatial orientations.

The implementation and solving of cubes with marked centres in cubes of size 3 to a much higher size has been the focus of most of the *Unravel* program upgrades since 2012. Solving marked cubes of size greater than 3 is more complex than solving standard cubes of similar size.

Cubie markings may impose a pixel penalty that limits the size of cube for which markings can be applied. Two forms of marking have been used for the *Unravel* program:

- Numerical marking: Uses a black coloured number graphic in range 1 to 4 superimposed on the standard cubie colour in the background.
- Corner marking: Uses a black coloured square a quarter the cubie size in area superimposed on the standard cubie colour in the background.

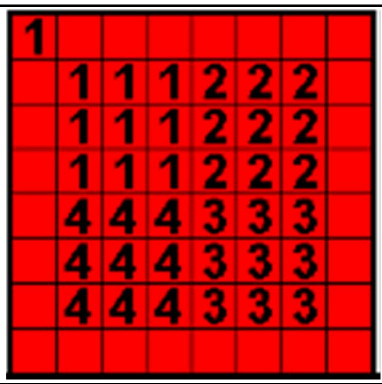
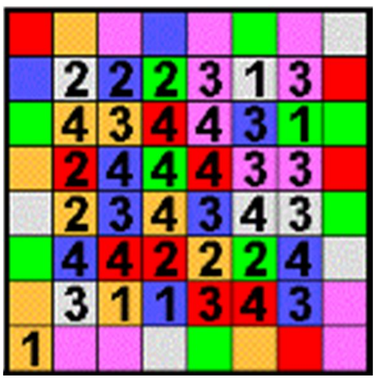
- There is a direct correspondence between numerical and corner marking. A top left corner quadrant marking is equivalent a numerical marking 1, second quadrant to 2, third quadrant to 3, and fourth quadrant to 4. The following table illustrates these different forms of marking.

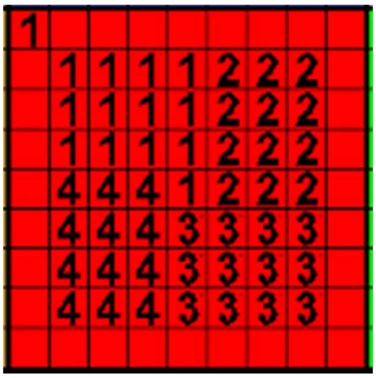
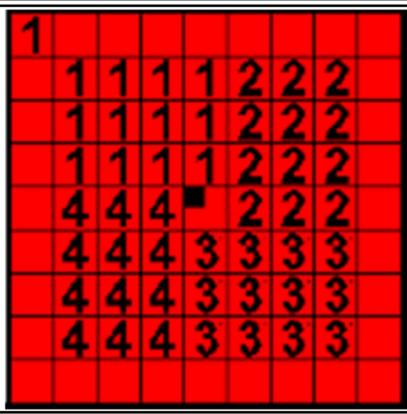
Marking	0 deg	90 deg	180 deg	270 deg
Numeric	1	2	3	4
Corner				

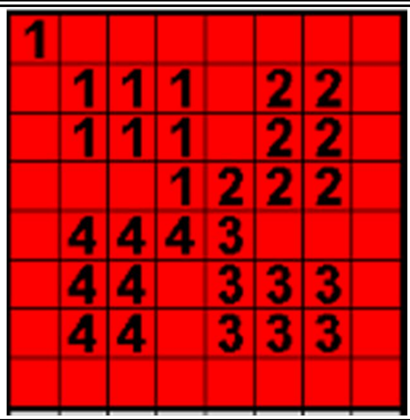
An upper cube size limit of about 32 applies for many monitors in common use when numerical marking is used. Corner marking (except for absolute centre cubies for cubes of odd size) is more difficult to use when unscrambling a cube than the numerical form but allows marking to be extended above the numerical marking limit.

Because transfer of cubies between orbits is impossible, the same 1-2-3-4 markings can be used for each orbit. With the exception of the absolute centre cubies for cubes of odd size, there are 24 centre cubies (4 per face) in each orbit. If n is cube size, there will be $((n - 2)^2 - a)/4$ orbits where a is zero if n is even or a is one if n is odd. Hence there are 9 orbits for a size 8 cube as shown below in images 1 and 2 for example.

For hardware cubes the cubie markings would change orientation as the face is rotated. However, with the exception of the absolute centre cubies for cubes of odd size orientation cannot be changed independently of position, so position defines orientation and vice versa. For software cubes (except for the absolute centre cubies for cubes of odd size) it is therefore necessary to change only position.

1		Front face for a size 8 cube in the set state.
2		An example of a face for a size 8 cube after scrambling.

3		Front face for a size 9 cube in the set state when a numerical graphic is used to mark the rotation of the absolute centre cubie.
4		An example of a face for a size 9 cube in the scrambled state when a numerical graphic is used to mark the rotation of the absolute centre cubie.
5		Front face for a size 9 cube in the set state when a corner graphic is used to mark the rotation of the absolute centre cubie.
6		The identical example of a face for a size 9 cube in the scrambled state when a corner graphic is used to mark the rotation of the absolute centre cubie.

7		Front face for a size 8 cube in the set state when this size cube is used to simulate absolute centre cubie rotation for a size 7 cube.
8		An example of a face for a size 8 cube after scrambling when this size cube is used to simulate absolute centre cubie rotation for a size 7 cube.

For cubes of odd size the absolute centre cubies are subject to restricted movement - they can only occupy 6 possible positions (one on each face) whereas other centre cubies can occupy 24 possible positions. To track their rotation, numerical marking as in images 3 and 4 or corner marking as in images 5 and 6 can be used. The default for marked cubes is to use corner marking for absolute centre cubies and numerical marking where possible for other centre cubies. When the system shown in images 3 to 6 is in use the absolute centre cubie images are subject to change or possible change for every quarter turn of the cube.

An earlier form of centre cubie marking that used a one-size-up cube to simulate absolute centre cubie marking is still available and is illustrated in images 7 and 8. With the one-up simulation the program does not permit any move that would allow relative movement between the two most central layers. In all other respects it behaves as a size 7 cube with central pairs of centre and edge cubies behaving as single cubies. For image 8 note that the 1-2-3-4 sequence is maintained but the central block has been rotated. For non-absolute-centre-cubies, numerical marking is omitted on the cubie on the clockwise side of the central pairs. Multiple images for the absolute centre cubies are not required for this option - the rotational status of the absolute centre cubie is defined by its cubie positions. The program does not allow the one-size-up marking option to be swapped with that used in images 3 to 6 for a partially unscrambled cube.

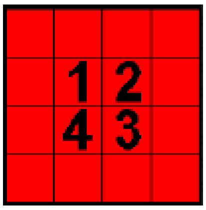
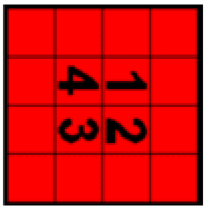
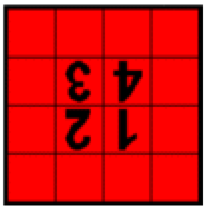
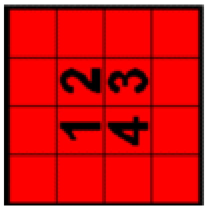
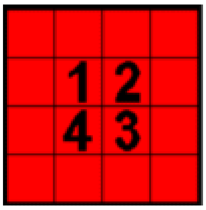
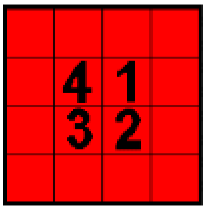
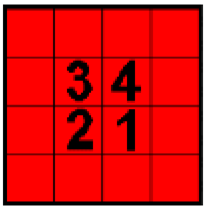
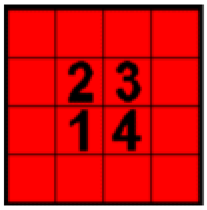
The selection of which of the marked absolute centre cubie rotation display options the user adopts is provided by options available when the *Unravel* program is run and the cube size to which the corner cubie style can extend the marking option above the numerical marking limit can also be changed when the program is run.

The full block of centre cubies can be rotated relative to the surrounding corner and edge cubies. The orientation of the block of centre cubies is given by the single number "1" marker on the left upper corner on each face for the set cube. The nearest centre cubie to that marker must have a "1" indicator for correct final alignment. Macros are available to correct for any such misalignment. However, if the user sets the white face at the bottom and the red face on the front and follows the approach to solving provided herein, the user will never experience this form of misalignment. Hence, although the availability of the marker is useful, it is not essential. If there were no marker and the user adopted a different orientation of the whole cube there would be 2048 possible arrangements for the solved cube that would appear correct.

Corner marking makes use of a simpler graphic than the numerical form. It allows the marking limit to be extended above the numerical marking limit (up to the cube size 99 program limit) but is less convenient to handle than the numerical form.

3.2 Application to Orbits Comprising 24 Centre Cubies

Cubes with centre cubie marking are a rarity. The usual approach (for the size 3 hardware cube) is to use images that overlap into the outer layer.[4] Outer layer marking might be helpful in some cases but would not be essential. Image marking was considered unnecessarily complex for the software implementation. For the *Unravel* program a simple 1-2-3-4 marking has been adopted. For the size 4 cube there is a single orbit of 24 centre cubies. Centre cubie movement on a single face is considered below for a hardware cube and for the *Unravel* software cube with the 1-2-3-4 marking.

Cube Type	0° (360°)	90° (-270°)	180° (-180°)	270° (-90°)
Hardware				
Software (*Unravel*)				

For hardware cubes, the orientation of the centre cubies changes with position as the face is rotated. For the *Unravel* software program, only the position changes as the face is rotated. For hardware cubes, if the position is known the orientation is known and vice versa. In that case it is impossible to change one without the other. It follows that restricting movement to position changes only as for the *Unravel* software program meets all necessary requirements. The 1-2-3-4 marking system can be used for hardware cubes but in that case it is not as convenient as when it is used for software cubes.

For cubes of size greater than 4, there will be multiple 24 centre cubie orbits. Since movement of cubies between orbits is impossible, the same 1-2-3-4 markings can be used for all orbits.

Since the author produced the first version of cubes with marked centres having an upper limit of size 16, the upper limit, when numerical marking graphics are in use, has been extended to size 32 typically. Cube numerical markings reduce the reachable upper limit relative to that applicable for cubes without marked centres. It was found that the centre cubie markings became distorted if the coloured numerical element was less than 8 pixel wide (9 pixel with a 1 pixel separator). However, that limit would provide acceptable viewing for most people. To achieve the size 32 limit it was found necessary to be able to scroll the bottom face partially or fully out of view for monitors with limited available vertical pixels. With the scroll option enabled, the 32 size limit was achievable for a wide range of notebook computers. The size limit could be increased beyond 32 for some larger monitors. When the *Unravel* program is run the numerical size limit is calculated and users can view the limit at run-time.

For a size n cube:

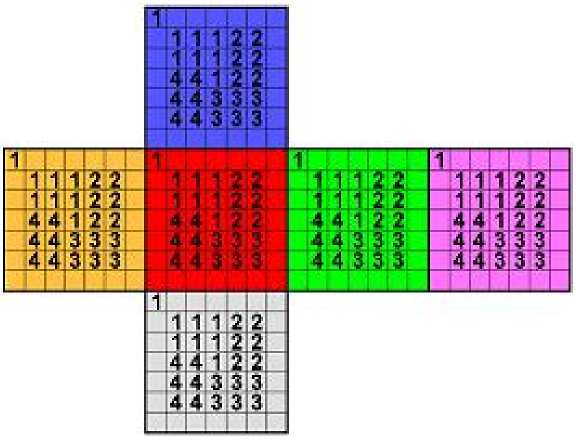
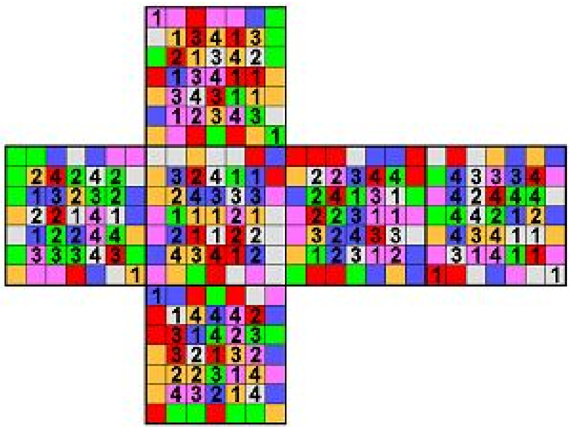
Parameter	Symbol	Value
Has value 0 if n is even, 1 if it is odd	a	n mod 2
Number of centre cubies per face	x	$(n-2)^2$
Number of 4 cubie sets in different orbits per face	y	$((n-2)^2 - a)/4$

Values of y as a function of n are given below:

n	4	5	6	7	8	9	10	11	12	13	14	15	16
y	1	2	4	6	9	12	16	20	25	30	36	42	49

Images 1 and 3 below show set cubes and images 2 and 4 show scrambled cubes for size 7 and 8 *Unravel* program cubes respectively. The way that the 1-2-3-4 sets are presented in the set cube for the *Unravel* application provides just one possibility. In theory, provided there is just one set of 1, 2, 3 and 4 in each orbit on each face then any arrangement is valid. The arrangement for the *Unravel* program was chosen with user-convenience in mind.

The initial set cube always appears with the Down face coloured white and the Front face coloured red. Although not necessary for the *Unravel* program, a simple outer layer “1” marker has been added. Such markers would probably be required for hardware cubes and some other software implementations. The absolute centre cubies for odd size cubes always carry markings (either numerical or corner type) for cubes considered in this section. One-size-up simulation of absolute centre cubie marking for cubes of odd size will be considered in the next section.

1. Set size 7 cube with numerical absolute centre cubie marking	2 Scrambled size 7 cube example with numerical absolute centre cubie marking
	

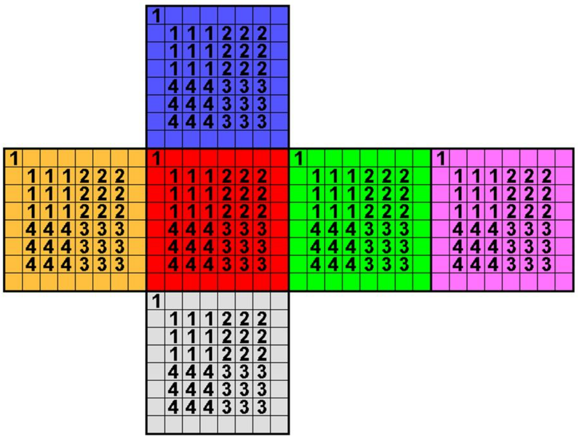
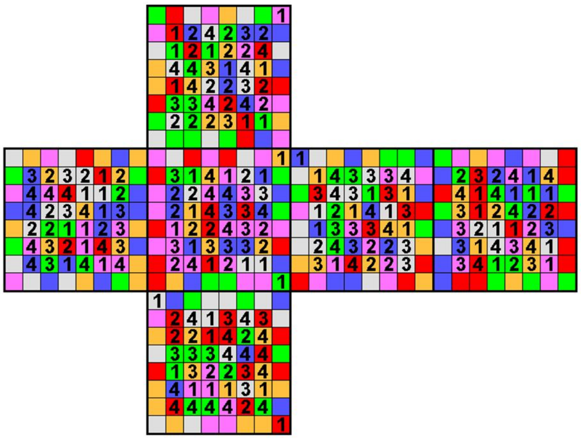
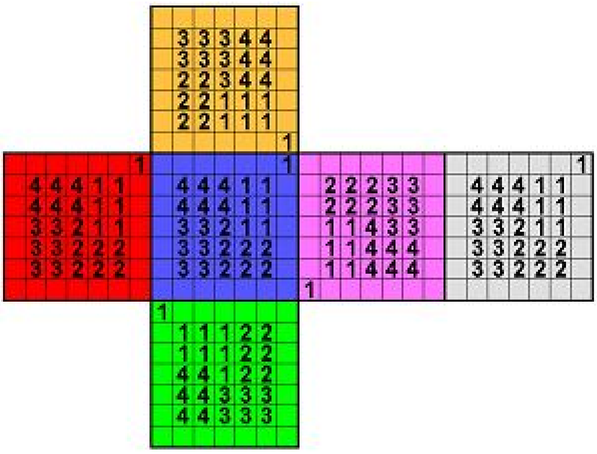
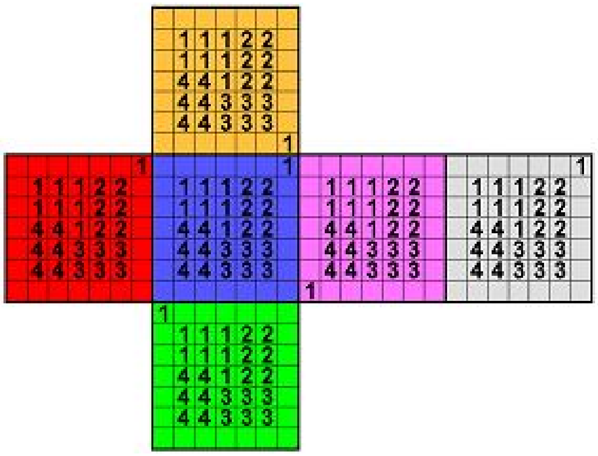
3. Set size 8 cube	4. Scrambled size 8 cube example
	

Image 5 shows a size 7 set cube in a different spatial orientation to that for the original set state. Since a different spatial orientation does not change cube state, such a state if arrived at first in the solving process would be considered a valid solution. Note that the ones are now not always in the top left corner but are always in the corner with the “1” outer layer marker. Trying to solve the cube other than to match the original spatial orientation would probably be overly cumbersome. Image 6 shows what results if the cube shown in image 5 has the centre blocks rearranged[3] so that the centre cubie “ones” are all in the top left of each face. Note the misalignment with the outer face marker. The arrangement of image 6 is not a valid solution. It is just one of the 2047 possibilities that does not represent a valid solution.

5. Set size 7 cube with new spatial orientation	6. Rearranged cube not the correct solution
	

The coordinates of each centre cubie can be defined. The coordinates will be represented as (row-number, column-number). Assume that the top left facelet has coordinates (1, 1) and hence the top left centre cubie has position (2, 2). In general, if a given cubie has coordinates (i, j), then the coordinates of the other locations on the same orbit for a size n cube are as given in the following table.

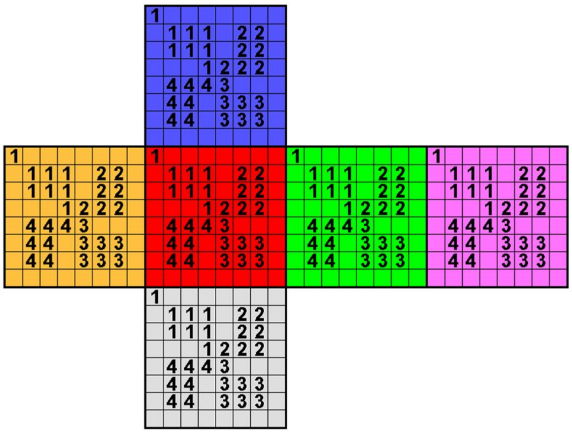
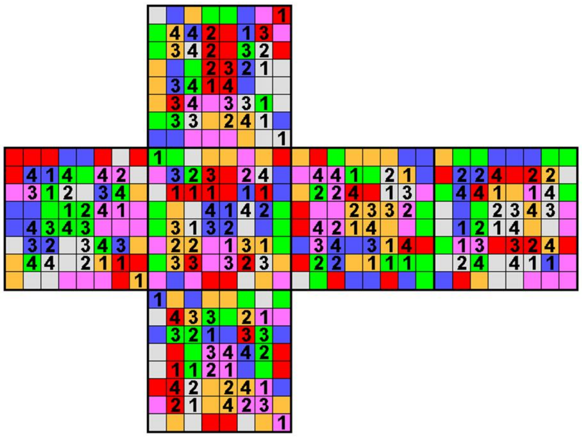
Reference	Reference + 90°	Reference + 180°	Reference - 90°
(i, j)	(j, n – i + 1)	(n – i + 1, n – j + 1)	(n – j + 1, i)

[3] Image 5 is realized by applying `WFWU` to the original set state (image 3) and the adjustment for mage 6 is realized by applying `-M20WUM20WU-M20WU-M20WU` to the image 5 state. The meaning of these moves is defined in Appendix A and Appendix C.

The coordinate numbers are unrelated to the numbers used for markings and apply equally to cubes with or without markings.

3.3 Application to Odd Size Cubes with Absolute Centre Orbit Comprising Six Cubies

Rotation of the absolute centre cubie for cubes of odd size can be simulated by using the next size up cube with some movement restrictions. The required movement restrictions are the prohibiting of any move that would allow any relative movement between the two central layers on any face in either direction. The set state is shown in image 7 and a scrambled arrangement is shown in image 8.

7. Set size 8 cube that simulates a size 7 cube that allows rotation of marked absolute centre cubies	8. Example of cube as shown to the left after scrambling
	

The simulated absolute centre cubie on each face has the size of four ordinary centre cubies but behaves as a single cubie. The 1-2-3-4 sequence is retained for the absolute centre cubies but that sequence can be rotated when the cube is scrambled as shown in image 8. Other centre cubies on the centre lines are shown with the clockwise side of each pair without any numerical marker. These pairs behave as single cubies. Likewise the central edge cubie pairs behave as single cubies.

4. Solving Rubik’s Family Cubes with Marked Centres

4.1 Comparison with Cubes Having Unmarked Centres

There is no such thing as a wrong way to solve a cube. There are many ways for solving cubes and there are many references to ways cubes with unmarked centres can be solved, particularly those up to size 5. The author describes one method[5] of solving cubes of any size with the usual style of unmarked centres. That method is based largely on the use of macros in the *Unravel* application.[3] Macros are applicable only to so ftware cubes that use text entry commands to change cube state. In the main, macros provide algorithm shortcuts and are responsible for providing a major means of simplifying the solving of software cubes.

The notation used in the following sections for basic and macro moves is defined in Appendix A. The *Unravel* application displays the six faces of the cube in a two-dimensional manner as illustrated below.

	U		
L	**F**	**R**	**B**
	D		

All algorithms for solving cubes with unmarked centres are also applicable to cubes with the alternative marked centres' style but some additional algorithms are needed to complete their solving. Algorithms for aligning marked centres for the standard size 3 cube are described by Monroe.[4] At the time of writing the original version of this document, the author was unaware of references (other than this document) that detail a way of solving cubes of size greater than 3 with marked centres. That may be a result of the unavailability of large cubes with marked centres at the time of writing, other than that provided by the author.[3]

4.2 Solving the 24 Centre Cubies in Each Marked Cube Orbit

Consideration is given in this section to the solving of the following cubes with marked centres:

- even size cubes of size 4 and above.
- odd size cubes of size 3 and above.

While it is possible to solve cubes using any spatial orientation, the choice of any orientation other than that shown for the set state (refer to image 1 and image 3 in the previous section) would add an unnecessary difficulty as illustrated in the image 5 example in Sec. 3.2 for a different spatial orientation. The solution described here will be based on a cube with bottom face white and front face red. In that case the "1" markings on centre cubies will always appear in the top left quadrant after proper alignment.

Alignment of centre cubies requires both cubie colour and cubie marking to be correct. It can be impossible to align the final layer centre cubies if all the edge cubies have not been fully aligned. There are still only 24!/2 possible arrangements with everything but the centre cubies fixed in position and orientation but inter-dependence between edge and centre cubies can mean it is not possible to align the final set of centre cubies without changing the alignment of the edge cubies. The impact of edge cubie misalignment on centre cubie placement possibilities is considered in further detail in another document[6] written by the author.

Furthermore, because the alignment of final layer edge cubies can upset some previous alignment of centre cubie markings, there is a further good reason for leaving the alignment of most centre cubies until all edge cubies have been placed. That approach, while different to that most commonly used for cubes with unmarked centres, will be adopted herein.

A step-by-step summary of the author's way of solving the cube is given in the following table: The 2-digit macros shown in the table apply only for cubes of size less than or equal to 16. Above that range 4-digit macros must be used and their form will be correctly shown in the on-line macro help.

Step	Description	Comment
1	Align white centres	This can be done either without or with macros (Appendix B). For odd size cubes the white face is the one in which the unmarked absolute white centre cubie resides. For even size cubes any face can be made the white face.
2	Align edge and corner cubies on white face	With the “1” centre cubies located at the upper left quadrant of the white face, the red face is on the front side, the green face s on the right side, the violet face is on the back side, and the orange face is on the left side. For the steps defined here, the white face alignment will not be disturbed by future steps.
3	Align remaining eight edge cubie sets	For cubes of odd size it is important to first align the absolute centre cubie if numerical or corner marking of that cubie is in use. If the non-preferred one-size-up simulation is in use, it is important to first align the four absolute centre cubies in the central section to match the edge cubie selection on the bottom face, as moving them later would be troublesome. Alignment here means all cubies in each set must have the same colour on corresponding surfaces (i.e. none in a given set needs to be flipped) and all except those for the final (blue) face are correctly positioned.
4	Align corner and edge cubies in the final (blue) face.	The moves used here are mainly those for a standard size 3 cube. For cubes of even size two edge sets may need to be swapped.
5	Align both colour and marking on centre cubies for red, green, violet and orange faces but just colour for the blue face.	The centre cubie alignment macros used for unmarked cubes are used here.
6	Align the markings for the final (blue) face.	At this point all centre cubies on the blue face will be of the correct colour but the numerical or corner markings in most orbits are likely to be incorrect. As indicated earlier, it is essential that all edge cubies are fully aligned before tackling this last step.

Macros (or algorithms) to assist with steps 1 to 5 in the above table are described elsewhere[5] and, in most cases, will not be further examined herein. Most attention will be devoted to step 6 which requires special moves that arise because of the presence of the markings.

At the completion of step 5, only the markings on the upper face will require correction. As use will be made of the centre cubie alignment macros of the form Mrc (refer to Appendix A), where *r* is the row number and *c* is the column number of the centre cubie on the front face that needs to be replaced by a cubie in the same position on the upper face. The macros are of the form:

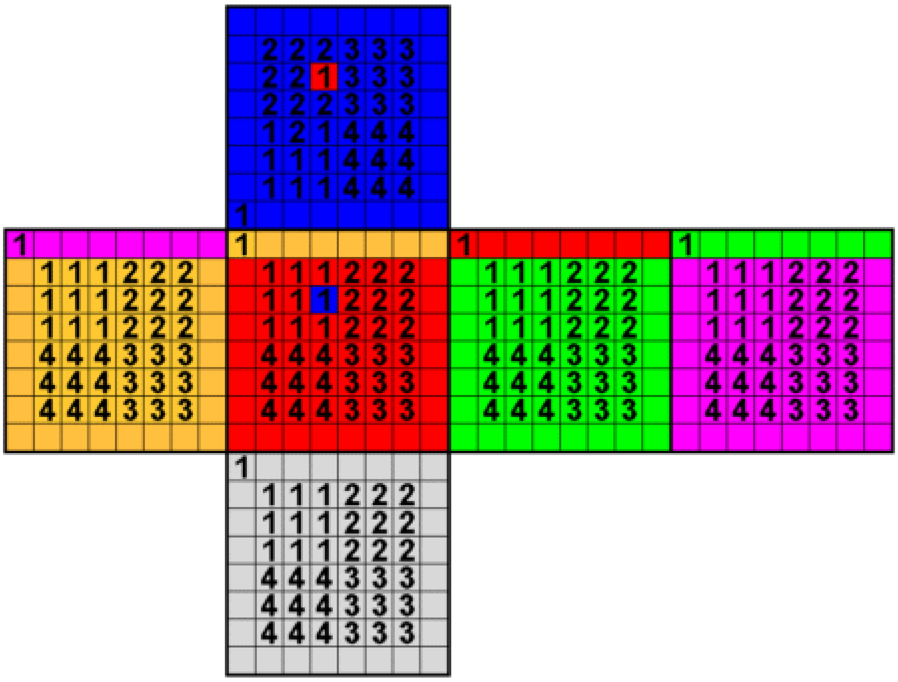
```
Mrc = #cB#rU-#cBU#cB-#rU-#cBUU
```

If the top row on the face is designated row 1 and the left column on the face is designated column 1, then M22 will be the top left centre macro designation corresponding to its position. For example, the macro for the (row 3, column 4) centre cubie alignment is:

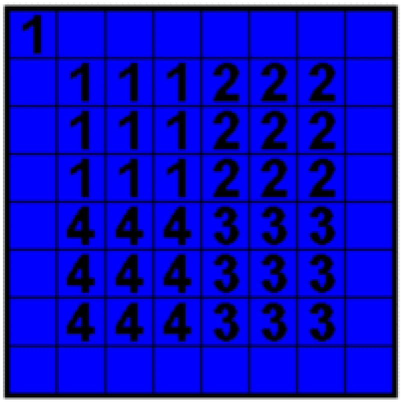
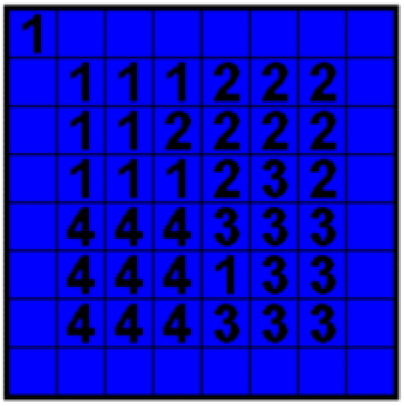
```
M34 = #4B#3U-#4BU#4B-#3U-#4BUU
```

For cubes of size greater than 16, four-digit numerical macros are used. Instead of M34 M0304 would be used in such cases.

The result from the application of the M34 macro to a set size 8 cube is shown below.



Consider the application of M34M34UU to a set cube.

Set upper face	After M34M34UU
	

The application of M34M34UU changes the 1-2-3-4 sequence to 2-3-1-4. In this case the position of 4 is unchanged whereas there is an anticlockwise re-positioning of 1-2-3. The reverse UU-M34-M34 will produce a clockwise re-positioning of 1-2-3.

Using M34M34, -M34-M34 and appropriate U movements in various sequences will allow correction of all possibilities. There is a degree of similarity to what is required to perform position alignment of edge cubie sets for the final layer. If all arrangements were reachable there would be 4! (equal to 24) possible arrangements for each orbit. However, if no change except within a given set of four centre cubies in a particular orbit is allowed, any change that would involve an odd number of swaps (odd parity) would be unreachable. Hence there are 4!/2 (equal to 12) possibilities. If “rc” represents the coordinates of the cubie in the top left quadrant for a given orbit, then the required correction algorithm (to change the sequence to the required 1-2-3-4) is as tabulated in the following table.

Sequence	Swaps	Parity	Reachable	Correction algorithm
1234	0	even	Yes	None required
1243	1	odd	No	
1324	1	odd	No	
1342	2	even	Yes	`U-Mrc-MrcU`
1423	2	even	Yes	`-UMrcMrc-U`
1432	1	odd	No	
2341	3	odd	No	
2314	2	even	Yes	`UU-Mrc-Mrc`
2413	3	odd	No	
2431	2	even	Yes	`-U-Mrc-Mrc-U`
2134	1	odd	No	
2143	2	even	Yes	`UUMrcMrcUMrcMrcU`
3124	2	even	Yes	`MrcMrcUU`
3142	3	odd	No	
3214	1	odd	No	
3241	2	even	Yes	`-Mrc-MrcUU`
3412	2	even	Yes	`MrcMrc-UMrcMrcU`
3421	3	odd	No	
4123	3	odd	No	
4132	2	even	Yes	`UMrcMrcU`
4213	2	even	Yes	`UUMrcMrc`
4231	1	odd	No	
4312	3	odd	No	
4321	2	even	Yes	`-UMrcMrcUMrcMrc`

If all the possibilities were to have macros defined, there would need to be 11 for every 4 already defined. That is outside the scope for the *Unravel* program that will limit the number of numerical macros to n^2 where n is cube size. Hence, when using the *Unravel* program, users will need to enter the algorithms as shown in the above table with appropriate coordinates substituted for “rc”. Fortunately, only one correction algorithm or none will be required for each orbit.

A simple way of finding the current sequence is to note the first number in the top left quadrant and then apply –U three times noting the number that appears at the same location each time. Make sure a final –U is applied to restore the original.

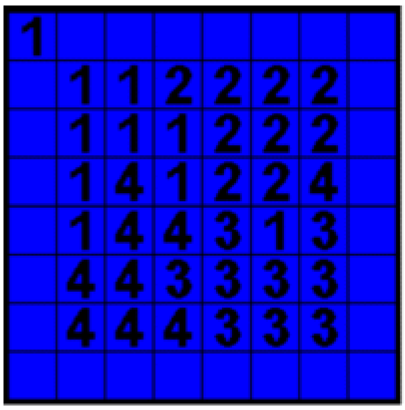
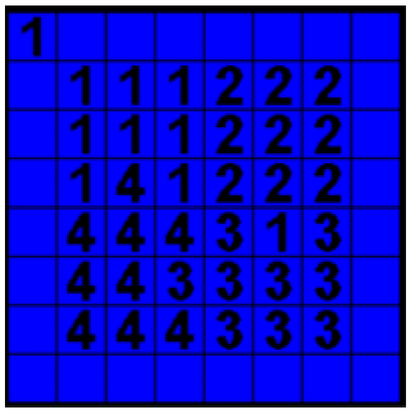
It may arise that the user finds that the final set of four centre cubies in a particular orbit appears to require an odd parity correction (1432 for example). There are two possible explanations:

1. Edge cubies have not been properly aligned before commencing this step. This is not likely to be the cause of the problem as misalignment of edge cubies is readily observable. However, if this is the cause of the problem no manipulation of the centre cubies will enable all the centre cubies on the final face to be aligned. Even parity for each 24 centre cubie orbit still applies and there are still 24!/2 possible centre cubie arrangements if no change of edge or corner cubies occurs. It just means that the 1234

sequence is not achievable until the edge cubie alignment is corrected. That correction should be completed before proceeding with any correction as detailed below in 2.

2. A more likely cause of the problem is that somewhere in the 24 centre cubie orbit there is a misalignment (eg. a 2 and 3 need to be swapped). Note that an even number of swaps must apply over the whole orbit. Such an error becomes more likely to occur as the cube size is increased. In general terms the odd parity correction can arise if the total number of swaps required in the 24 cubie orbit is odd. If there is an even number of erroneous swaps required external to the final layer then the final layer can be fully aligned but automatic detection of the solved state (as for the *Unravel* program) will fail.

An example of the application of the above algorithms is illustrated below. On the first image seven of the nine orbits have correct alignment. The two remaining orbits need attention. The first one that needs alignment has upper left quadrant coordinates (2,4) and the second (4,3). The first requires the sequence 2431 to be converted to 1234 and the second requires 4213 to be converted to 1234. From the above table, algorithm `-U-M24-M24-U` will perform the first correction as illustrated in the second image. Algorithm UUM43M43 will perform the final alignment.

Two orbits need correction	One orbit needs correction
	

While it is probably unlikely that a centre cubie block (comprising all the centre cubies on the given face) will become misaligned relative to the surrounding edge and corner cubies, the following universal macros (based on those defined by Monroe[4] for the size 3 cube and defined in the *Unravel* program) can be used to restore the blocks to their correct attitude if that were ever required. Macro M20 (same as <20>) rotates the centre cubie block on the front face a quarter turn clockwise and the centre cubie block on the upper face a quarter turn counter-clockwise. Macro M30 rotates the centre cubie block on the upper face a half turn.

```
M20 = -WBB-F-WUU-DWB-BFU-WBB-FWU-UDWB-BF-U
M30 = URLUU-R-LURLUU-R-L
```

All moves (macros or algorithms) that are defined for use in the solving of a marked size 8 cube are given in Appendix C.

4.3 Solving the Marked Absolute Centres for Cubes of Odd Size

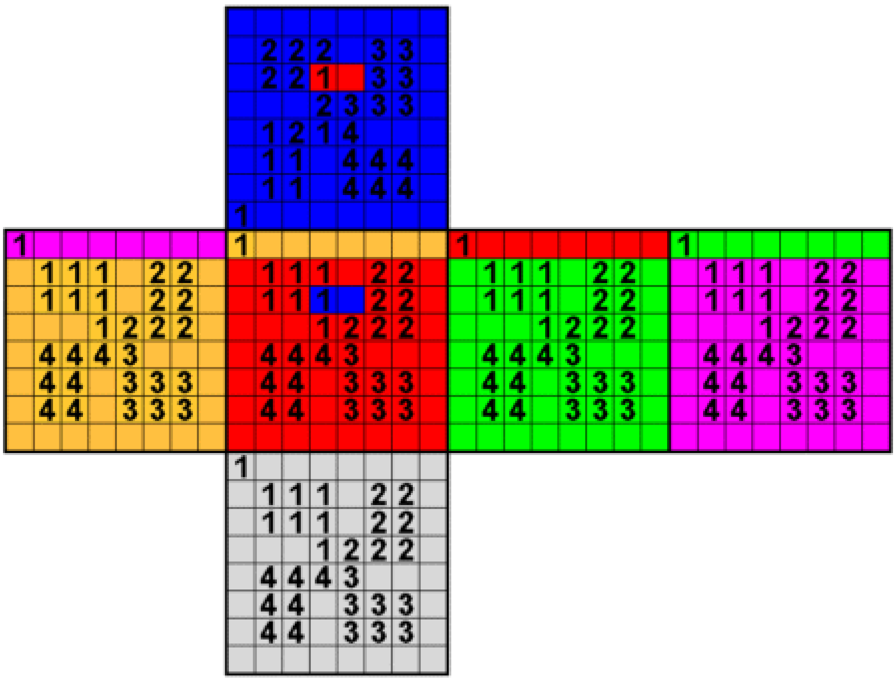
This section considers the rotation and alignment of the six absolute centre cubies in cubes of odd size. To achieve this, the default method is to mark the absolute centre cubie rotation using a numerical or corner graphic form for cubes of size up to about 32 and the corner graphic any size above that range. Full details of applicable macros are provided in Appendix D for a size 7 cube example.

An alternative one=size-up cube with rotation restrictions is available as a non-preferred option. Any quarter turn that would result in relative movement between the pair of central rows or columns is inhibited. For example, for a size 8 cube (simulating a size 7 cube with marked absolute centres), the following movements about the upper face axis are excluded: 4U, #4U and #5U. Similar movements about other face axes are also excluded. To move the central pair of rows about the upper face axis use `5U-3U`. The size 7 cube simulation behaves as if the cube has the same number of cubies as a standard size 7 cube. The size of the simulated absolute centre cubie is 4 units (where one unit is the size of an ordinary centre cubie) and the other cubies on the centre lines have a size of 2 units. Full detailsof the macros or algoriths applicable to the one-size-up size 7 simulation is prvided in Appendix E.

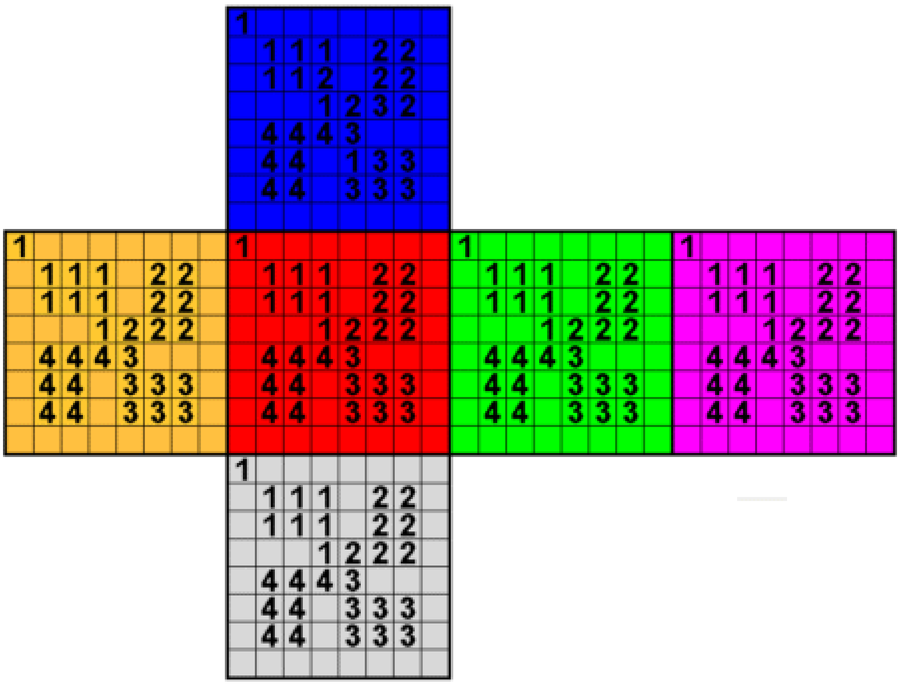
The absolute centre cubie (represented by four marked cubies) on each face behaves as a single cubie that can be rotated as a block. Other cubies on row and column centre lines can only be moved as pairs. This requires changed centre cubie alignment macros. No macros are defined with coordinates matching the unmarked centre cubies in the central pairs (refer to images 7 and 8 in Section 3.3). All *Unravel* macros for a cube size 7 example are reproduced in Appendix D. For that size cube the following macros shift locked pairs of centre cubies:

```
M24 = -3B5B#2U3B-5BU-3B5B-#2U3B-5BUU
M34 = -3B5B#3U3B-5BU-3B5B-#3U3B-5BUU
M46 = #6B-3U5U-#6BU#6B3U-5U-#6BUU
M47 = #7B-3U5U-#7BU#7B3U-5U-#7BUU
M52 = #2B-3U5U-#2BU#2B3U-5U-#2BUU
M53 = #3B-3U5U-#3BU#3B3U-5U-#3BUU
M65 = -3B5B#6U3B-5BU-3B5B-#6U3B-5BUU
M75 = -3B5B#7U3B-5BU-3B5B-#7U3B-5BUU
```

An example of a central pair that needs to be set on the front face for a size 7 cube simulation is shown below.



After applying M34UU the following front face correction results.



First quadrant macros (M24 and M34 for the size 8 cube) would be used in algorithms, defined in the previous section, for final alignment of central pairs in the upper face.

The alignment of the absolute centre cubie is a straight forward extension of that required for a marked size 3 centre cubie (as provided by Monroe[4]). Two macros are defined for this purpose. Macro M00 (same as <00>) is used to rotate the central four cubies on the front face a quarter turn clockwise and the central four cubies on the up face a quarter turn counter-clockwise. Macro M10 is used to rotate the central four cubies on the up face a half turn. The definitions of M00 and M10 vary according to cube size. The alignment of the central four cubies on the various faces can be done either before or after the other centre cubies are aligned.

If $x = (n/2 - 1)$ and $y = (x + 2)$ where n is the simulation cube size, the general form of these macros will be:

```
M00 = xB-yBxU-yU-xByBUxB-yB-xUyU-xByB-U
M10 = xUxRxLxUxU-xR-xLxUxRxLxUxU-xR-xL
```

For the size 3 cube simulation the value of x will equal 1 and in that case omit the 1.

For the size 7 cube (with marked absolute centres simulated using a size 8 cube), the required macros are:

```
M00 = 3B-5B3U-5U-3B5BU3B-5B-3U5U-3B5B-U
M10 = 3U3R3L3U3U-3R-3L3U3R3L3U3U-3R-3L
```

4.4 Centre Cubie Block Transfer Macro

For the author’s preferred method for marked cubes of sequentially aligning all centre cubies on a face before doing the same on the next face, the ability to transfer all centre cubies, irrespective of facelet colour, from the front to the upper face can be useful, particularly for cubes of large size and for the early stage of the centre cubie alignment process (up to the fourth face only).

The macro that performs the transfer involves a rearrangement of the cubies on these faces and that is of no consequence as this transfer would precede alignment using the standard Centre Cubie Location Macros.[5] Except for the centres cubies on the front and upper faces, no change in the alignment of any other cubie occurs. No transfer of the absolute centre cubies for odd size cubes occurs unless absolute centre cubie rotation simulation is active.

For cube size 4 macro <13> performs the transfer and for all cubes of larger size macro <40> (or <0400> for cube of size greater than 16) does the transfer. These cube names are the same for the optional origins [0,0] and [1,1]. For cubes of size greater than 4 the transfer macro makes use of one auxiliary macro for cubes of even size and three auxiliary macros for cubes of odd size.

For the size 4 cube only one transfer macro as defined below for origin [1,1] applies. Replace <22> with <11> when origin [0,0] applies.

```
* <13> = <22>UFU<22>FU<22>FUU<22>-UF
```

For cubes of size greater than 4, the transfer and auxiliary macros defined below are the same for the [0,0] and [1,1] origins.

<u>Even cubes of size 6 and above</u>

The auxiliary macro names vary with cube size. Define <0a> as the single auxiliary macro required for cubes of even size. In that case for a cube of size n, a = n/2. Define b = n – 1. For cubes of size greater than 16, 4-digit macros apply and in such cases replace 0a with 00aa where aa = n/2 and any leading zero is preserved.

```
* <0a> = -BbBaU-UB-bBU-BbB-aUUB-bBUU
* <40> = <0a>U<0a>FFUU<0a>U<0a>-UFFU
```

For a size 16 cube:

```
* <08> = -B15B8U-UB-15BU-B15B-8UUB-15BUU
* <40> = <08>U<08>FFUU<08>U<08>-UFFU
```

For a size 32 cube:

```
* <0016> = -B31B16U-UB-31BU-B31B-16UUB-31BUU
* <0400> = <0016>U<0016>FFUU<0016>U<0016>-UFFU
```

<u>Odd cubes of size 5 and above without absolute centre cubie rotation simulation</u>

The auxiliary macro names vary with cube size. Define <0a>, <0b> and <0c> as the three auxiliary macros required for cubes of odd size. In that case for a cube of size n:

a = (n - 1)/2, b = a + 1, c = a + 2, and d = n - 1.

For cubes of size greater than 16, 4-digit macros apply and in such cases replace 0a with 00aa where aa = (n - 1)/2 and any leading zero is preserved.

```
* <0a> = -BdBaU-UB-dBU-BdB-aUUB-dB-U
* <0b> = <0a>U<0a>FF<0a><0a)>FF-U
* <0c> = -BaB#bUB-aBU-BaB-#bUB-aB-U
* <40> = <0b>U<0c>FFUU<0c>UUFF-U
```

For size 15

```
* <07> = -B14B7U-UB-14BU-B14B-7UUB-14B-U
* <08> = <07>U<07>FF<07><07>FF-U
```

```
* <09> = -B7B#8UB-7BU-B7B-#8UB-7B-U
* <40> = <08>U<09>FFUU<09>UUFF-U
```

For size 31

```
* <0015> = -B30B15U-UB-30BU-B30B-15UUB-30B-U
* <0016> = <0015>U<0015>FF<0015><0015>FF-U
* <0017> = -B15B#16UB-15BU-B15B-#16UB-15B-U
* <0400> = <0016>U<0017>FFUU<0017>UUFF-U
```

Odd cubes of size 5 and above with absolute centre cubie rotation simulation

The auxiliary macro names vary with cube size. Define <0a>, <0b> and <0c> as the three auxiliary macros required for cubes of odd size. In that case for a cube of size (n – 1) which is simulated using a cube of size n:

a = n/2, b = a + 1, c = a + 2, d = n – 1, and e = a - 1.

For cubes of size greater than 16, 4-digit macros apply and in such cases replace 0a with 00aa where aa = n/2 and any leading zero is preserved.

```
* <0a> = -BdBbU-UB-dBU-BdB-bUUB-dB-U
* <0b> = <0a>U<0a>FF<0a><0a)>FF-U
* <0c> = -BaB#bUB-aBU-BaB-#bUB-aB-U
* <0A> = -eBbBeU-UeB-bBU-eBbB-eUUeB-bB
* <40> = <0b>FUU<0c>U-F
```

For size 15 (with size 16 simulation)

```
* <08> = -B15B9U-UB-15BU-B15B-9UUB-15B-U
* <09> = <08>U<08>FF<08><08>FF-U
* <0A> = -7B9B7U-U7B-9BU-7B9B-7UU7B-9B
* <40> = <09>FUU<0A>U-F
```

For size 31 (with size 32 simulation)

```
* <0016> = -B31B17U-UB-31BU-B31B-17UUB-31B-U
* <0017> = <0016>U<0016>FF<0016><0016>FF-U
* <0018> = -15B17B15U-U15B-17BU-15B17B-15UU15B-17B
* <0400> = <0017>FUU<0018>U-F
```

References

1. Fraser, K., Rubik's Cube Extended: Derivation of Number of States for Cubes of Any Size and Values for up to Size 25x25x25. http://www.kenblackbox.com/cube/maths/cubestates.pdf
2. Fraser, K., Rules for Rubik's Family Cubes of All Sizes, http://www.kenblackbox.com/cube/maths/cuberules.pdf
3. Fraser, K., UnravelJ - Size 2x2x2 to 99x99x99 for standard cube, size 3x3x3 to 32x32x32 for centres with numerical markings and up to 99x99x99 for centres with corner markings, 2D, Java applet, Java Web Start or Java archive direct download, all platforms. http://www.kenblackbox.com/unravelrun.htm
4. Monroe, M., How to handle pictures or logos on the faces. http://www.alchemistmatt.com/cube/rubikcenter.html
5. Fraser, K., Instructions for Solving Rubik Family Cubes of Any Size, http://www.kenblackbox/cube/solving/cubesolving.pdf
6. Fraser, K., *Rules for Rubik's Family Cubes of All Sizes*, http://www.kenblackbox.com/cube/maths/cuberules.pdf.

Appendix A

Basic Move and Macro Command Notation

The six basic commands F (front), B (back), L (left), R (right), U (up) and D (down) produce a 90 degree clockwise (CW) rotation of the outer layer about the selected face axis. If a basic command is preceded by a minus sign (-F etc.) a 90 degree counter-clockwise (CCW) rotation of the outer layer about the selected face axis is produced.

For cubes of size greater than 3, the ability to rotate inner layers is essential. Firstly, a multiple layer command is defined as illustrated in the following example. 3R rotates outer three layers 90 degrees CW about the R face axis while -3R rotates outer three layers 90 degrees CCW about the R face axis. Double-digit prefixes (e.g. 12R) are accepted for cubes of large size. The prefix "W" can be used to rotate the whole cube. For instance, WU rotates the whole cube 90 degrees CW about the U face axis and similarly -WU produces a CCW rotation. For an *n*-layer cube the value "*n*" is internally substituted for W. W rotations don't change the state of the cube; they merely provide a different spatial orientation of the faces (putting the right face in the front for example).

A single inner layer rotation command is also defined as illustrated in the following example. #3R or 03R rotates only the third layer 90 degrees CW about the R face axis and similarly -#3R or -03R performs a CCW rotation. Note that #3R = 3R-2R and -#3R = -3R2R.

Optional single layer rotation symbol 0 (zero) in lieu of #, would normally be used for keyboard entry. In that way there is no need to use the shift key (lower case alphabetical characters are always internally converted and displayed as upper case). Symbol # improves readability when used in help text.

The term "macro" as used in computer science is a rule or pattern that specifies how a certain input sequence should be mapped into an output sequence. Macros are normally used to map a short string to a longer string (sequence of instructions). Macros simplify things in *Unravel* by providing short-cuts for long sets (or frequently used short sets) of instructions on how to rotate the cube. In Rubik's cube parlance, command sequences that perform non-trivial transformations are often referred to as algorithms. Such algorithms are prime candidates for macro implementation.

A macro can be called by any alphabetical character except F, B, L, R, U, D, W and M (reserved symbol defined below). There are thus 18 allowable alphabetical macro symbols:

A, C, E, G, H, I, J, K, N, O, P, Q, S, T, V, X, Y, and Z.

A macro can also be called by a number enclosed within angle brackets (or without brackets if the character M precedes the number). Such macros are referred to as *numerical* macros. Numerical macros conform to the arrangement <*(Row)(Column)*> where *Row* and *Column* conform to a coordinate structure that facilitates the use of such macros particularly for centre cubie placement. For convenience, designate a row digit as *r* and a column digit as *c*. For cubes with marked centres (size less than 17), a total of two digits is used. That results in numerical macros of the form <*rc*>. Macro nesting to any depth is permitted.

Optional numerical macro symbol M in lieu of angle brackets (i.e. M*rc* in lieu of <*rc*> would normally be used for keyboard entry. In that way there is no need to use the shift key (lower case alphabetical characters are always internally converted and displayed as upper case). Angle brackets improve readability when used in help text.

Appendix B

Using Macros to Align Centre Cubies on First Face

For cubes with marked centres it is recommended that the white face be the first face to have its centre cubies aligned. The following steps can be used if macros are used.

1. Place the nominated white face at the front. For cubes of odd size the absolute centre cubie defines the white face. For odd size cubes using absolute centre cubie simulation, rotate front face so that 1 appears at top left (but it can be simply changed later if this action is missed). For cubes of even size, the white centre is not defined until the user selects any face as the white face. The white centre is then defined as the intersection of the vertical and horizontal centre lines on the front face.
2. Transfer any incorrectly positioned white cubies on the front face to the upper face.
3. Transfer white cubies on upper face to correct locations on front face.
4. Use WU to transfer the right side to the front and transfer white cubies on that face to the upper face (the Centre Cubie Block Transfer Macro will facilitate this). Repeat steps 3.
5. Repeat 4 for the remaining two faces on the middle section.
6. Use move FFWFWF to move the bottom set of centre cubies to the upper face.
7. Transfer the remaining white centre cubies from the upper face to the white face.
8. Use WL to place the white face at the bottom.

Appendix C

Algorithms or Macros for Use with Marked Size 8 Cube

1. Section of Cube to be Set Without Macros

The extent to which macros are used is optional and users may decide to not use some or all of them. When macros are used, there are some restrictions on the way the cube is unscrambled.

Choose white as the bottom face. With red chosen as the front face, set the 1-2-3-4 sequence such that the "1" centre cubie(s) on the bottom face appear(s) at the top left. Align bottom layer edge cubies to conform to this selection. Although placement macros are provided for all centre cubies except those on the bottom layer, the placement of the centre cubies on the lower half of the F, R, B and L faces without using macros may be preferred. Also, the bottom layer centre cubies can be placed using centre cubie macros by initially doing the placement on the front face.

2. Centre Cubie Block Transfer Macro

For users, who sequentially align all centre cubies on a face before doing the same on the next face, the ability to transfer all centre cubies, irrespective of facelet colour, from the front to the upper face can be useful, particularly for cubes of large size and for the early stage of the centre cubie alignment process (up to the fourth face only).

Macro <40> transfers all centre cubies on the front face for this even size cube. The transfer involves a rearrangement of the cubies on these faces and that is of no consequence as this transfer would precede alignment using macros listed in the "Centre Cubie Location Macros" section. Except for the centres cubies on the front and upper faces, no change in the alignment of any other cubie occurs.

```
* <40> = <04>U<04>FFUU<04>U<04>-UFFU
```

Auxiliary macro <04> is used in the definition of macro <40>.

```
* <04> = -B7B4U-UB-7BU-B7B-4UUB-7BUU
```

3. Centre Cubie Location Macros

Alignment macros are provided for all the centre cubies on the front face, where rows are counted from the top (Row 1) downwards.

Centre cubies can be aligned in any order. No change in the alignment of edge or corner cubies applies when the centre cubie location macros are used (so users can align edge cubies before centre cubies if they wish). Centre cubie macros swap an upper layer cubie with one in the equivalent position on the front layer. No other movement occurs on the front layer but some rearrangement occurs on the upper layer in conformity with an even parity requirement. Centre cubie alignment is a two step process. Firstly, if desired centre cubies are not located on the upper face, a face containing the required centre cubies is placed at the front and centre cubie macros are used to transfer them to the upper face. Secondly, the face being aligned is then placed in the front and centre cubie macros are used to retrieve the required centre cubies that have just been placed on the upper face.

The details are given below.

```
Row 01:  xx   xx  ....  xx   xx
Row 02:  xx  <22> .... <27>  xx
Row 03:  xx  <32> .... <37>  xx
Row 04:  xx  <42> .... <47>  xx
Row 05:  xx  <52> .... <57>  xx
Row 06:  xx  <62> .... <67>  xx
Row 07:  xx  <72> .... <77>  xx
Row 08:  xx   xx  ....  xx   xx
```

xx = no centre cubie macro applicable

```
* <22> = #2B#2U-#2BU#2B-#2U-#2BUU
* <23> = #3B#2U-#3BU#3B-#2U-#3BUU
* <24> = #4B#2U-#4BU#4B-#2U-#4BUU
* <25> = #5B#2U-#5BU#5B-#2U-#5BUU
* <26> = #6B#2U-#6BU#6B-#2U-#6BUU
* <27> = #7B#2U-#7BU#7B-#2U-#7BUU
* <32> = #2B#3U-#2BU#2B-#3U-#2BUU
* <33> = #3B#3U-#3BU#3B-#3U-#3BUU
* <34> = #4B#3U-#4BU#4B-#3U-#4BUU
* <35> = #5B#3U-#5BU#5B-#3U-#5BUU
* <36> = #6B#3U-#6BU#6B-#3U-#6BUU
* <37> = #7B#3U-#7BU#7B-#3U-#7BUU
* <42> = #2B#4U-#2BU#2B-#4U-#2BUU
* <43> = #3B#4U-#3BU#3B-#4U-#3BUU
* <44> = #4B#4U-#4BU#4B-#4U-#4BUU
* <45> = #5B#4U-#5BU#5B-#4U-#5BUU
* <46> = #6B#4U-#6BU#6B-#4U-#6BUU
* <47> = #7B#4U-#7BU#7B-#4U-#7BUU
* <52> = #2B#5U-#2BU#2B-#5U-#2BUU
* <53> = #3B#5U-#3BU#3B-#5U-#3BUU
* <54> = #4B#5U-#4BU#4B-#5U-#4BUU
* <55> = #5B#5U-#5BU#5B-#5U-#5BUU
* <56> = #6B#5U-#6BU#6B-#5U-#6BUU
* <57> = #7B#5U-#7BU#7B-#5U-#7BUU
* <62> = #2B#6U-#2BU#2B-#6U-#2BUU
* <63> = #3B#6U-#3BU#3B-#6U-#3BUU
* <64> = #4B#6U-#4BU#4B-#6U-#4BUU
* <65> = #5B#6U-#5BU#5B-#6U-#5BUU
* <66> = #6B#6U-#6BU#6B-#6U-#6BUU
* <67> = #7B#6U-#7BU#7B-#6U-#7BUU
* <72> = #2B#7U-#2BU#2B-#7U-#2BUU
* <73> = #3B#7U-#3BU#3B-#7U-#3BUU
* <74> = #4B#7U-#4BU#4B-#7U-#4BUU
* <75> = #5B#7U-#5BU#5B-#7U-#5BUU
* <76> = #6B#7U-#6BU#6B-#7U-#6BUU
* <77> = #7B#7U-#7BU#7B-#7U-#7BUU
```

Auxiliary macro "K", which provides a shorthand means for rotating all but the upper layer, is likely to be applied many times when using the above macros to align centre cubies.

```
* K = WDU
```

4. First Layer Corner Location Macros

Locate first layer at the bottom and shift from top left to bottom left. Use macro "A" if the colour of the front face of the upper cubie is the same as that for the bottom face of the cube, "C" if the colour of the side face of the upper cubie is the same as that for the bottom face of the cube, and "E" if the colour of the top face of the upper cubie is the same as that for the bottom face of the cube.

```
* A = FU-F
* C = -L-UL
* E = -LULUUC
```

5. Edge Cubie Adjustment Macros

Locate final layer at top. Use "P" to exchange left side front edge set with right side top edge set such that top face relocates on left face. Use "-P" to exchange left side front edge set with top front edge set such that top face relocates on front face. To obtain matched edge sets for transfer to and from the top layer, use 2U to 7U rotations.

```
* P = FUURU-RUU-F
```

Use "Q" to invert set of edge cubies located at the left side of the front face.

```
* Q = P-UP
```

6. Final Edge Set Alignment Macros

Some adjustment of the final edge set may be required before all the final layer edge cubies achieve the same sense. Place the final edge cubie set to be aligned at the front edge of the upper layer. Adjustment can proceed in any order until all cubies in the final edge set are aligned in the same sense.

```
<14> converts [- - b b - -] to [- - a a - -].
<13> converts [- b - - b -] to [- a - - a -].
<12> converts [b - - - - b] to [a - - - - a].
```

Note: "a" and "b" refer to the two faces of the edge cubies.
"-" means "don't care" (ie. either "a" or "b").

```
* <12> = #2R#2RBBUU#2LUU-#2RUU#2RUUFF#2RFF-#2LBB#2R#2R
* <13> = #3R#3RBBUU#3LUU-#3RUU#3RUUFF#3RFF-#3LBB#3R#3R
* <14> = #4R#4RBBUU#4LUU-#4RUU#4RUUFF#4RFF-#4LBB#4R#4R
```

7. Single Edge Set Flip Macro for Final Layer Edge Cubies

Before proceeding further with the final layer alignment it is essential that the number of edge sets that need to be flipped is not odd (one or three). If an odd number of matched edge sets need to be flipped on the final layer (can only occur for even size cubes), use "S" to flip the set located at the front of the upper layer. Alternatively, the required result can be achieved using the "Final Edge Set Alignment Macro(s)".

```
* S = 4R4RRRBBUU4L-LUU-4RRUU4R-RUUFF4R-RFF-4LLBB4R4RRR
```

8. Final Layer Corner Location Macros

Locate final layer at top and use "T" to swap the two front corners.

```
* T = B-U-FU-B-UFUU
```

Sequences containing T, U and -U can be used to cover all other final layer corner location possibilities not covered by the use of a single T macro. Such sequences together with alternative macro definitions are as follows:

With lower right corner cubie of upper face fixed, use "UUTUTU" or "G" to provide CW movement of three remaining corner cubies.

With lower right corner cubie of upper face fixed, use "-UT-UTUU" or "-G" to provide CCW movement of three remaining corner cubies.

Use "TUT-UT" or "H" to provide diagonal swap of lower left with upper right corner cubie.

* G = UUTUTU
* H = TUT-UT

9. Final Layer Edge Cubie Location Macros

Locate final layer at top. Use "V" to obtain a CW movement, and "-V" to provide a CCW movement, of the leftmost three edge cubie sets.

```
* V = LLU-FBLL-BFULL
```

Sequences containing V, U and -U can be used to cover other final layer edge cubie location possibilities not covered by the use of a single V or -V macro. Such sequences together with alternative macro definitions are as follows:

Swap left with right, and up with down edge cubies (or cubie sets) using "V-UVU" or "I".

Swap left with down, and up with right cubies (or cubie sets) using "-V-U-VU" or "J".

```
* I = V-UVU
* J = -V-U-VU
```

Just one opposite pair of edge cubie sets may need to be swapped. Refer to "Final Layer Diametrically Opposite Edge Set Swap Macro" if that applies.

10. Final Layer Diametrically Opposite Edge Set Swap Macro

Locate final layer at top. Locate one of the sets to be swapped at the front and use "X" to exchange it with the edge set at the back.

```
* X = 4R4RRRUU4R4RRRUU4U4UUU4R4RRR4U4UUU
```

11. Final Layer Corner Twirl Macro

Locate final layer at the top. Use "Y" to provide a CW twirl of the front right corner cubie. Locate another corner cubie at the front right corner and use "-Y" to provide a CCW twirl of that cubie. Up to two applications of this macro may be required to obtain correct orientation of all corner cubies.

```
* Y = -RDRFD-F
```

12. Final Layer Edge Set Flip Macro

Locate final layer at the top. Use "Z" to flip edge set located at the front. Locate another top layer edge set requiring a flip at the right side and use "-Z". Up to two applications of this macro may be required to obtain correct orientation of all edge sets.

```
* Z = FU-DLLUUDDR
```

13. Alignment of Marked Centre Cubies in the Upper Layer

Solving the cube will be simplified if the spatial orientation adopted conforms with that for the original set state. In that case the white face will be at the bottom and the red face will be at the front, and the "one" corner will appear at the top left for each face for the solved cube. Although rotating the whole cube to a different position does not change its state, it can make sorting out the correct numerical sequences a nightmare.

At this point all edge and corner cubies and all marked centre cubies in all but the upper layer are correctly aligned. The blue colour of all the centre cubies on the upper face is correct but the numerical sequences for the four centre cubies in each orbit may be incorrect. The correct alignment is 1-2-3-4 clockwise where the red is the front face and centre cubie number one is in the upper left position.

If there were no movement restrictions on the last face to be aligned there would be 24 (i.e. 4!) possible arrangements for each set-of-four cubies. However, only those that involve an even number of swaps (even parity) are reachable. That reduces the reachable possibilities to 12. Only one of the 12 is correct. To accommodate all possibilities, 11 algorithms would be required for each set-of-four centre cubies.

Correction algorithms using functions of the previously defined "Centre Cubie Location Macros" for the top left member of the set-of-four can be used. The required algorithms are listed below where rc represents the (row,column) coordinates expressed in hexadecimal format. To accommodate all possibilities with macros would require almost three times the number used for the "Centre Cubie Location Macros" that cover all centre cubies on the front face. That would exceed the two numerical character limit (256 macros). Hence macros are not provided. However, for any given set-of-four centre cubies either no algorithm or just one algorithm application would be required. It is recommended that the user determines the sequence that needs correction and then copy the relevant algorithm given below (including the rc variable) to the Command Sequence Input dialog and substitute the appropriate rc values.

Sequence	Swaps	Parity	Correction algorithm
1234	0	even	None required
1243	1	odd	
1324	1	odd	
1342	2	even	`U-Mrc-MrcU`
1423	2	even	`-UMrcMrc-U`
1432	1	odd	
2341	3	odd	
2314	2	even	`UU-Mrc-Mrc`
2413	3	odd	
2431	2	even	`-U-Mrc-Mrc-U`
2134	1	odd	
2143	2	even	`UUMrcMrcUMrcMrcU`
3124	2	even	`MrcMrcUU`
3142	3	odd	
3214	1	odd	
3241	2	even	`-Mrc-MrcUU`
3412	2	even	`MrcMrc-UMrcMrcU`
3421	3	odd	
4123	3	odd	
4132	2	even	`UMrcMrcU`
4213	2	even	`UUMrcMrc`
4231	1	odd	
4312	3	odd	
4321	2	even	`-UMrcMrcUMrcMrc`

While it is probably unlikely that the centre cubie block will become misaligned relative to the surrounding edge and corner cubies, the following universal macros can be used to restore the blocks to their correct attitude if that were ever required. Macro M20 (same as <20>) rotates the centre cubie block on the front face a quarter turn clockwise and the centre cubie block on the upper face a quarter turn counter-clockwise (the reverse occurs for -M20). Macro M30 rotates the centre cubie block on the upper face a half turn. WU means rotate the whole cube about the U face and similarly for other faces.

```
* <20> = -WBB-F-WUU-DWB-BFU-WBB-FWU-UDWB-BF-U
* <30> = URLUU-R-LURLUU-R-L
```

Appendix D

Algorithms or Macros for Use with Marked Size 7 Cube with Preferred Absolute Centre Cubie Rotation

1 Section of Cube to be Set Without Macros

The extent to which macros are used is optional and users may decide to not use some or all of them. When macros are used, there are some restrictions on the way the cube is unscrambled.

Choose white as the bottom face. With red chosen as the front face, set the 1-2-3-4 sequence such that the "1" centre cubie(s) on the bottom face appear(s) at the top left. Align bottom layer edge cubies to conform to this selection. Although placement macros are provided for all centre cubies except those on the bottom layer, the placement of the centre cubies on the lower half of the F, R, B and L faces without using macros may be preferred. Also, the bottom layer centre cubies can be placed using centre cubie macros by initially doing the placement on the front face.

2. Centre Cubie Block Transfer Macro

For users, who sequentially align all centre cubies on a face before doing the same on the next face, the ability to transfer all centre cubies, irrespective of facelet colour, from the front to the upper face can be useful, particularly for cubes of large size and for the early stage of the centre cubie alignment process (up to the fourth face only).

Macro <40> transfers all centre cubies on the front face except for the absolute centre cubie for this odd size cube. The transfer involves a rearrangement of the cubies on these faces and that is of no consequence as this transfer would precede alignment using macros listed in the "Centre Cubie Location Macros" section. Except for the centres cubies on the front and upper faces, no change in the alignment of any other cubie occurs.

```
* <40> = <04>U<05>FFUU<05>UUFF-U
```

```
Auxiliary macros <03>, <04> and <05> are used in the definition
of macro <40>.
```

```
* <03> = -B6B3U-UB-6BU-B6B-3UUB-6B-U
* <04> = <03>U<03>FF<03><03>FF-U
* <05> = -B3B#4UB-3BU-B3B-#4UB-3B-U
```

3. Centre Cubie Location Macros

Alignment macros are provided for all the centre cubies on the front face, where rows are counted from the top (Row 1) downwards.

Centre cubies can be aligned in any order. No change in the alignment of edge or corner cubies applies when the centre cubie location macros are used (so users can align edge cubies before centre cubies if they wish). Centre cubie macros swap an upper layer cubie with one in the equivalent position on the front layer. No other movement occurs on the front layer but some rearrangement occurs on the upper layer in conformity with an even parity requirement. Centre cubie alignment is a two step process. Firstly, if desired centre cubies are not located on the upper face, a face containing the required centre cubies is placed at the front and centre cubie macros are used to transfer them to the upper face. Secondly, the face being aligned is then placed in the front and centre cubie macros are used to retrieve the required centre cubies that have just been placed on the upper face.

The details are given below.

```
Row 01:  xx   xx  ....  xx  ....  xx   xx
Row 02:  xx  <22> .... <24> .... <26>  xx
Row 03:  xx  <32> .... <34> .... <36>  xx
Row 04:  xx  <42> ....  xx  .... <46>  xx
Row 05:  xx  <52> .... <54> .... <56>  xx
Row 06:  xx  <62> .... <64> .... <66>  xx
Row 07:  xx   xx  ....  xx  ....  xx   xx
```

xx = no centre cubie macro applicable

```
* <22> = #2B#2U-#2BU#2B-#2U-#2BUU
* <23> = #3B#2U-#3BU#3B-#2U-#3BUU
* <24> = #4B#2U-#4BU#4B-#2U-#4BUU
* <25> = #5B#2U-#5BU#5B-#2U-#5BUU
* <26> = #6B#2U-#6BU#6B-#2U-#6BUU
* <32> = #2B#3U-#2BU#2B-#3U-#2BUU
* <33> = #3B#3U-#3BU#3B-#3U-#3BUU
* <34> = #4B#3U-#4BU#4B-#3U-#4BUU
* <35> = #5B#3U-#5BU#5B-#3U-#5BUU
* <36> = #6B#3U-#6BU#6B-#3U-#6BUU
* <42> = #2B#4U-#2BU#2B-#4U-#2BUU
* <43> = #3B#4U-#3BU#3B-#4U-#3BUU
* <45> = #5B#4U-#5BU#5B-#4U-#5BUU
* <46> = #6B#4U-#6BU#6B-#4U-#6BUU
* <52> = #2B#5U-#2BU#2B-#5U-#2BUU
* <53> = #3B#5U-#3BU#3B-#5U-#3BUU
* <54> = #4B#5U-#4BU#4B-#5U-#4BUU
* <55> = #5B#5U-#5BU#5B-#5U-#5BUU
* <56> = #6B#5U-#6BU#6B-#5U-#6BUU
* <62> = #2B#6U-#2BU#2B-#6U-#2BUU
* <63> = #3B#6U-#3BU#3B-#6U-#3BUU
* <64> = #4B#6U-#4BU#4B-#6U-#4BUU
* <65> = #5B#6U-#5BU#5B-#6U-#5BUU
* <66> = #6B#6U-#6BU#6B-#6U-#6BUU
```

Auxiliary macro "K", which provides a shorthand means for rotating all but the upper layer, is likely to be applied many times when using the above macros to align centre cubies.

```
* K = WDU
```

4. First Layer Corner Location Macros

Locate first layer at the bottom and shift from top left to bottom left. Use macro "A" if the colour of the front face of the upper cubie is the same as that for the bottom face of the cube, "C" if the colour of the side face of the upper cubie is the same as that for the bottom face of the cube, and "E" if the colour of the top face of the upper cubie is the same as that for the bottom face of the cube.

```
* A = FU-F
* C = -L-UL
* E = -LULUUC
```

5. Edge Cubie Adjustment Macros

Locate final layer at top. Use "P" to exchange left side front edge set with right side top edge set such that top face relocates on left face. Use "-P" to exchange left side front edge set with top front edge set such that top face relocates on front face. To obtain matched edge sets for transfer to and from the top layer, use 2U to 6U rotations.

```
* P = FUURU-RUU-F
```

Use "Q" to invert set of edge cubies located at the left side of the front face.

```
* Q = P-UP
```

6. Final Edge Set Alignment Macros

Some adjustment of the final edge set may be required before all the final layer edge cubies achieve the same sense. Place the final edge cubie set to be aligned at the front edge of the upper layer. Adjustment can proceed in any order until all cubies in the final edge set are aligned in the same sense. No adjustment of the central edge cubie for odd size cubes is provided by these macros. As this is an odd size cube, alignment should proceed so as to match the central edge cubie.

```
<13> converts [- b a b -] to [- a a a -].
<12> converts [b - a - b] to [a - a - a].
```

Note: "a" and "b" refer to the two faces of the edge cubies.
"-" means "don't care" (ie. either "a" or "b").

```
* <12> = #2R#2RBBUU#2LUU-#2RUU#2RUUFF#2RFF-#2LBB#2R#2R
* <13> = #3R#3RBBUU#3LUU-#3RUU#3RUUFF#3RFF-#3LBB#3R#3R
```

7. Final Layer Corner Location Macros

Locate final layer at top and use "T" to swap the two front corners.

```
* T = B-U-FU-B-UFUU
```

Sequences containing T, U and -U can be used to cover all other final layer corner location possibilities not covered by the use of a single T macro. Such sequences together with alternative macro definitions are as follows:

With lower right corner cubie of upper face fixed, use "`UUTUTU`" or "`G`" to provide CW movement of three remaining corner cubies.

With lower right corner cubie of upper face fixed, use "`-UT-UTUU`" or "`-G`" to provide CCW movement of three remaining corner cubies.

Use "`TUT-UT`" or "`H`" to provide diagonal swap of lower left with upper right corner cubie.

* G = UUTUTU
* H = TUT-UT

8. Final Layer Edge Cubie Location Macros

Locate final layer at top. Use "V" to obtain a CW movement, and "-V" to provide a CCW movement, of the leftmost three edge cubie sets.

* V = LLU-FBLL-BFULL

Sequences containing V, U and -U can be used to cover other final layer edge cubie location possibilities not covered by the use of a single V or -V macro. Such sequences together with alternative macro definitions are as follows:

Swap left with right, and up with down edge cubies (or cubie sets) using "V-UVU" or "I".

Swap left with down, and up with right cubies (or cubie sets) using "-V-U-VU" or "J".

* I = V-UVU
* J = -V-U-VU

9. Final Layer Corner Twirl Macro

Locate final layer at the top. Use "Y" to provide a CW twirl of the front right corner cubie. Locate another corner cubie at the front right corner and use "-Y" to provide a CCW twirl of that cubie. Up to two applications of this macro may be required to obtain correct orientation of all corner cubies.

* Y = -RDRFD-F

10. Final Layer Edge Set Flip Macro

Locate final layer at the top. Use "Z" to flip edge set located at the front. Locate another top layer edge set requiring a flip at the right side and use "-Z". Up to two applications of this macro may be required to obtain correct orientation of all edge sets.

* Z = FU-DLLUUDDR

11. Alignment of Marked Absolute Centre Cubie for Odd Size Cubes

Use M00 (same as <00>) to rotate absolute centre cubie on the front face a quarter turn clockwise and the absolute centre cubie on the up face a quarter turn counter-clockwise. Use M10 to rotate the absolute centre cubie on the up face a half turn. The alignment of the absolute centre cubie on the various faces can be done either before or after the other centre cubies are aligned. Alignment of all centre cubies in the final layer except the absolute centre ones can be done using the methods detailed in the "Alignment of Marked Centre Cubies in the Upper Layer" section.

* <00> = -#4B-#4U#4BU-#4B#4U#4B-U
* <10> = 3U3R3L3U3U-3R-3L3U3R3L3U3U-3R-3L

12. Alignment of Marked Centre Cubies in the Upper Layer

Solving the cube will be simplified if the spatial orientation adopted conforms with that for the original set state. In that case the white face will be at the bottom and the red face will be at the front, and the "one" corner will appear at the top left for each face for the solved cube. Although rotating the whole cube to a different position does not change its state, it can make sorting out the correct numerical sequences a nightmare.

At this point all edge and corner cubies and all marked centre cubies in all but the upper layer are correctly aligned. The blue colour of all the centre cubies on the upper face is correct but the numerical sequences for the four centre cubies in each orbit may be incorrect. The correct alignment is 1-2-3-4 clockwise where the red is the front face and centre cubie number one is in the upper left position.

If there were no movement restrictions on the last face to be aligned there would be 24 (i.e. 4!) possible arrangements for each set-of-four cubies. However, only those that involve an even number of swaps (even parity) are reachable. That reduces the reachable possibilities to 12. Only one of the 12 is correct. To accommodate all possibilities, 11 algorithms would be required for each set-of-four centre cubies.

Correction algorithms using functions of the previously defined "Centre Cubie Location Macros" for the top left member of the set-of-four can be used. The required algorithms are listed below where rc represents the (row,column) coordinates expressed in hexadecimal format. To accommodate all possibilities with macros would require almost three times the number used for the "Centre Cubie Location Macros" that cover all centre cubies on the front face. That would exceed the two numerical character limit (256 macros). Hence macros are not provided. However, for any given set-of-four centre cubies either no algorithm or just one algorithm application would be required. It is recommended that the user determine the sequence that needs correction and then copy the relevant algorithm given below (including the rc variable) to the Command Sequence Input dialog and substitute the appropriate rc values.

Sequence	Swaps	Parity	Correction algorithm
1234	0	even	None required
1243	1	odd	
1324	1	odd	
1342	2	even	`U-Mrc-MrcU`
1423	2	even	`-UMrcMrc-U`
1432	1	odd	
2341	3	odd	
2314	2	even	`UU-Mrc-Mrc`
2413	3	odd	
2431	2	even	`-U-Mrc-Mrc-U`
2134	1	odd	
2143	2	even	`-U-Mrc-Mrc-U-Mrc-MrcUU`
3124	2	even	`MrcMrcUU`
3142	3	odd	
3214	1	odd	
3241	2	even	`-Mrc-MrcUU`
3412	2	even	`-U-Mrc-MrcU-Mrc-Mrc`
3421	3	odd	
4123	3	odd	
4132	2	even	`UMrcMrcU`
4213	2	even	`-Mrc-MrcUU-Mrc-MrcUU`
4231	1	odd	
4312	3	odd	
4321	2	even	`-Mrc-Mrc-U-Mrc-MrcU`

While it is probably unlikely that the centre cubie block will become misaligned relative to the surrounding edge and corner cubies, the following universal macros can be used to restore the blocks to their correct attitude if that were ever required. Macro `M20` (same as `<20>`) rotates the centre cubie block on the front face a quarter turn clockwise and the centre cubie block on the upper face a quarter turn counter-clockwise (the reverse occurs for `-M20`). Macro `M30` rotates the centre cubie block on the upper face a half turn. `WU` means rotate the whole cube about the U face and similarly for other faces.

```
* <20> = -WBB-F-WUU-DWB-BFU-WBB-FWU-UDWB-BF-U
* <30> = URLUU-R-LURLUU-R-L
```

Appendix E

Algorithms or Macros for Use with Marked Size 7 Cube with Simulated Absolute Centre Cubie Rotation

1. Section of Cube to be Set Without Macros

The extent to which macros are used is optional and users may decide to not use some or all of them. When macros are used, there are some restrictions on the way the cube is unscrambled.

Choose white as the bottom face. With red chosen as the front face, set the 1-2-3-4 sequence such that the "1" centre cubie(s) on the bottom face appear(s) at the top left. Align bottom layer edge cubies to conform to this selection. Although placement macros are provided for all centre cubies except those on the bottom layer, the placement of the centre cubies on the lower half of the F, R, B and L faces without using macros may be preferred. Also, the bottom layer centre cubies can be placed using centre cubie macros by initially doing the placement on the front face.

2. Centre Cubie Block Transfer Macro

For users, who sequentially align all centre cubies on a face before doing the same on the next face, the ability to transfer all centre cubies, irrespective of facelet colour, from the front to the upper face can be useful, particularly for cubes of large size and for the early stage of the centre cubie alignment process (up to the fourth face only).

Macro <40> transfers all centre cubies on the front face except for the absolute centre cubie block for this special marked simulation of an odd size cube. The transfer involves a rearrangement of the cubies on these faces and that is of no consequence as this transfer would precede alignment using macros listed in the "Centre Cubie Location Macros" section. Except for the centres cubies on the front and upper faces, no change in the alignment of any other cubie occurs.

```
* <40> = <05>FUU<06>U-F
```

Auxiliary macros `<04>`, `<05>` and `<06>` are used in the definition of macro `<40>`.

```
* <04> = -B7B5U-UB-7BU-B7B-5UUB-7B-U
* <05> = <04>U<04>FF<04><04>FF-U
* <06> = -3B5B3U-U3B-5BU-3B5B-3UU3B-5B
```

3. Centre Cubie Location Macros

Alignment macros are provided for all the centre cubies on the front face, where rows are counted from the top (Row 1) downwards. When the rotation of a marked absolute centre cubie is to be simulated, the same centre cubie macros as those for the standard arrangement are used except for those on each side of the vertical and horizontal centre lines. Except for the four most central centre cubies, special macros that move the cubies in pairs relative to these centre lines are used. Cubies on the clockwise side of these pairs have no numerical identification and no macros with their coordinates are defined. Alignment of the pairs is achieved using macros with coordinates of the pairs which have numerical marking.

Centre cubies can be aligned in any order. No change in the alignment of edge or corner cubies applies when the centre cubie location macros are used (so users can align edge cubies before centre cubies if they wish). Centre cubie macros swap an upper layer cubie with one in the equivalent position on the front layer. No other movement occurs on the front layer but some rearrangement occurs on the upper layer in conformity with an even parity requirement. Centre cubie alignment is a two step process. Firstly, if desired centre cubies are not located on the upper face, a face containing the required centre cubies is placed at the front and centre cubie macros are used to transfer them to the upper face. Secondly, the face being aligned is then placed in the front and centre cubie macros are used to retrieve the required centre cubies that have just been placed on the upper face.

The details are given below.

```
Row 01:  xx   xx  ....  xx   xx  ....  xx   xx
Row 02:  xx  <22> .... <24>  xx  .... <27>  xx
Row 03:  xx  <32> .... <34>  xx  .... <37>  xx
Row 04:  xx   xx  ....  xx   xx  .... <47>  xx
Row 05:  xx  <52> ....  xx   xx  ....  xx   xx
Row 06:  xx  <62> ....  xx  <65> .... <67>  xx
Row 07:  xx  <72> ....  xx  <75> .... <77>  xx
Row 08:  xx   xx  ....  xx   xx  ....  xx   xx
```

xx = no centre cubie macro applicable

```
* <22> = #2B#2U-#2BU#2B-#2U-#2BUU
* <23> = #3B#2U-#3BU#3B-#2U-#3BUU
* <24> = -3B5B#2U3B-5BU-3B5B-#2U3B-5BUU
* <26> = #6B#2U-#6BU#6B-#2U-#6BUU
* <27> = #7B#2U-#7BU#7B-#2U-#7BUU
* <32> = #2B#3U-#2BU#2B-#3U-#2BUU
* <33> = #3B#3U-#3BU#3B-#3U-#3BUU
* <34> = -3B5B#3U3B-5BU-3B5B-#3U3B-5BUU
* <36> = #6B#3U-#6BU#6B-#3U-#6BUU
* <37> = #7B#3U-#7BU#7B-#3U-#7BUU
* <46> = #6B-3U5U-#6BU#6B3U-5U-#6BUU
* <47> = #7B-3U5U-#7BU#7B3U-5U-#7BUU
* <52> = #2B-3U5U-#2BU#2B3U-5U-#2BUU
* <53> = #3B-3U5U-#3BU#3B3U-5U-#3BUU
* <62> = #2B#6U-#2BU#2B-#6U-#2BUU
* <63> = #3B#6U-#3BU#3B-#6U-#3BUU
* <65> = -3B5B#6U3B-5BU-3B5B-#6U3B-5BUU
* <66> = #6B#6U-#6BU#6B-#6U-#6BUU
* <67> = #7B#6U-#7BU#7B-#6U-#7BUU
* <72> = #2B#7U-#2BU#2B-#7U-#2BUU
* <73> = #3B#7U-#3BU#3B-#7U-#3BUU
* <75> = -3B5B#7U3B-5BU-3B5B-#7U3B-5BUU
* <76> = #6B#7U-#6BU#6B-#7U-#6BUU
* <77> = #7B#7U-#7BU#7B-#7U-#7BUU
```

Auxiliary macro "K", which provides a shorthand means for rotating all but the upper layer, is likely to be applied many times when using the above macros to align centre cubies.

```
* K = WDU
```

4. First Layer Corner Location Macros

Locate first layer at the bottom and shift from top left to bottom left. Use macro "A" if the colour of the front face of the upper cubie is the same as that for the bottom face of the cube, "C" if the colour of the side face of the upper cubie is the same as that for the bottom face of the cube, and "E" if the colour of the top face of the upper cubie is the same as that for the bottom face of the cube.

```
* A = FU-F
* C = -L-UL
* E = -LULUUC
```

5. Edge Cubie Adjustment Macros

Locate final layer at top. Use "P" to exchange left side front edge set with right side top edge set such that top face relocates on left face. Use "-P" to exchange left side front edge set with top front edge set such that top face relocates on front face. To obtain matched edge sets for transfer to and from the top layer, use 2U to 7U rotations.

```
* P = FUURU-RUU-F
```

Use "Q" to invert set of edge cubies located at the left side of the front face.

```
* Q = P-UP
```

6. Final Edge Set Alignment Macros

Some adjustment of the final edge set may be required before all the final layer edge cubies achieve the same sense. Place the final edge cubie set to be aligned at the front edge of the upper layer. Adjustment can proceed in any order until all cubies in the final edge set are aligned in the same sense.

```
<13> converts [- b a a b -] to [- a a a a -].
<12> converts [b - a a - b] to [a - a a - a].
```

Note: "a" and "b" refer to the two faces of the edge cubies.
"-" means "don't care" (ie. either "a" or "b").

```
* <12> = #2R#2RBBUU#2LUU-#2RUU#2RUUFF#2RFF-#2LBB#2R#2R
* <13> = #3R#3RBBUU#3LUU-#3RUU#3RUUFF#3RFF-#3LBB#3R#3R
```

7. Final Layer Corner Location Macros

Locate final layer at top and use "T" to swap the two front corners.

```
* T = B-U-FU-B-UFUU
```

Sequences containing T, U and -U can be used to cover all other final layer corner location possibilities not covered by the use of a single T macro. Such sequences together with alternative macro definitions are as follows:

With lower right corner cubie of upper face fixed, use "UUTUTU" or "G" to provide CW movement of three remaining corner cubies.

With lower right corner cubie of upper face fixed, use "-UT-UTUU" or "-G" to provide CCW movement of three remaining corner cubies.

Use "TUT-UT" or "H" to provide diagonal swap of lower left with upper right corner cubie.

```
* G = UUTUTU
* H = TUT-UT
```

8. Final Layer Edge Cubie Location Macros

Locate final layer at top. Use "V" to obtain a CW movement, and "-V" to provide a CCW movement, of the leftmost three edge cubie sets.

```
* V = LLU-FBLL-BFULL
```

Sequences containing V, U and -U can be used to cover other final layer edge cubie location possibilities not covered by the use of a single V or -V macro. Such sequences together with alternative macro definitions are as follows:

Swap left with right, and up with down edge cubies (or cubie sets) using "V-UVU" or "I".

Swap left with down, and up with right cubies (or cubie sets) using "-V-U-VU" or "J".

```
* I = V-UVU
* J = -V-U-VU
```

Just one opposite pair of edge cubie sets may need to be swapped. Refer to "Final Layer Diametrically Opposite Edge Set Swap Macro" if that applies.

9. Final Layer Corner Twirl Macro

Locate final layer at the top. Use "Y" to provide a CW twirl of the front right corner cubie. Locate another corner cubie at the front right corner and use "-Y" to provide a CCW twirl of that cubie. Up to two applications of this macro may be required to obtain correct orientation of all corner cubies.

```
* Y = -RDRFD-F
```

10. Final Layer Edge Set Flip Macro

Locate final layer at the top. Use "Z" to flip edge set located at the front. Locate another top layer edge set requiring a flip at the right side and use "-Z". Up to two applications of this macro may be required to obtain correct orientation of all edge sets.

```
* Z = FU-DLLUUDDR
```

11. Alignment of Marked Absolute Centre Cubie for Odd Size Cubes

This alignment uses the next higher cube size to simulate a single absolute centre cubie using a set-of-four centre cubies. In this case the 1-2-3-4 sequence for these centre cubies always remains true although the whole sequence can be rotated relative to that which applies for a set cube. Use M00 (same as <00> to rotate central four cubies on the front face a quarter turn clockwise and the central four cubies on the up face a quarter turn counter-clockwise. Use M10 to rotate the central four cubies on the up face a half turn. The alignment of the central four cubies on the various faces can be done either before or after the other centre cubies are aligned. Alignment of all centre cubies in the final layer except the central four can be done using the methods detailed in the "Alignment of Marked Centre Cubies in the Upper Layer" section.

```
* <00> = 3B-5B3U-5U-3B5BU3B-5B-3U5U-3B5B-U
* <10> = 3U3R3L3U3U-3R-3L3U3R3L3U3U-3R-3L
```

12. Alignment of Marked Centre Cubies in the Upper Layer

Solving the cube will be simplified if the spatial orientation adopted conforms with that for the original set state. In that case the white face will be at the bottom and the red face will be at the front, and the "one" corner will appear at the top left for each face for the solved cube. Although rotating the whole cube to a different position does not change its state, it can make sorting out the correct numerical sequences a nightmare.

At this point all edge and corner cubies and all marked centre cubies in all but the upper layer are correctly aligned. The blue colour of all the centre cubies on the upper face is correct but the numerical sequences for the four centre cubies in each orbit may be incorrect. The correct alignment is 1-2-3-4 clockwise where the red is the front face and centre cubie number one is in the upper left position.

If there were no movement restrictions on the last face to be aligned there would be 24 (i.e. 4!) possible arrangements for each set-of-four cubies. However, only those that involve an even number of swaps (even parity) are reachable. That reduces the reachable possibilities to 12. Only one of the 12 is correct. To accommodate all possibilities, 11 algorithms would be required for each set-of-four centre cubies.

Correction algorithms using functions of the previously defined "Centre Cubie Location Macros" for the top left member of the set-of-four can be used. The required algorithms are listed below where rc represents the (row,column) coordinates expressed in hexadecimal format. To accommodate all possibilities with macros would require almost three times the number used for the "Centre Cubie Location Macros" that cover all centre cubies on the front face. That would exceed the two numerical character limit (256 macros). Hence macros are not provided. However, for any given set-of-four centre cubies either no algorithm or just one algorithm application would be required. It is recommended that the user determine the sequence that needs correction and then copy the relevant algorithm given below (including the rc variable) to the Command Sequence Input dialog and substitute the appropriate rc values.

Sequence	Swaps	Parity	Correction algorithm
1234	0	even	None required
1243	1	odd	
1324	1	odd	
1342	2	even	`U-Mrc-MrcU`
1423	2	even	`-UMrcMrc-U`
1432	1	odd	
2341	3	odd	
2314	2	even	`UU-Mrc-Mrc`
2413	3	odd	
2431	2	even	`-U-Mrc-Mrc-U`
2134	1	odd	
2143	2	even	`UUMrcMrcUMrcMrcU`
3124	2	even	`MrcMrcUU`
3142	3	odd	
3214	1	odd	
3241	2	even	`-Mrc-MrcUU`
3412	2	even	`MrcMrc-UMrcMrcU`
3421	3	odd	
4123	3	odd	
4132	2	even	`UMrcMrcU`
4213	2	even	`UUMrcMrc`
4231	1	odd	
4312	3	odd	
4321	2	even	`-UMrcMrcUMrcMrc`

While it is probably unlikely that the centre cubie block will become misaligned relative to the surrounding edge and corner cubies, the following universal macros can be used to restore the blocks to their correct attitude if that were ever required. Macro M20 (same as <20>) rotates the centre cubie block on the front face a quarter turn clockwise and the centre cubie block on the upper face a quarter turn counter-clockwise (the reverse occurs for -M20). Macro M30 rotates the centre cubie block on the upper face a half turn. WU means rotate the whole cube about the U face and similarly for other faces.

```
* <20> = -WBB-F-WUU-DWB-BFU-WBB-FWU-UDWB-BF-U
* <30> = URLUU-R-LURLUU-R-L
```