TOTU : HP48GX PROGRAM FOR ANY CODING ON A THK CORDAGE ROUTE (single strand knots)

All along this document I have supposed that :
* **you own a HP48GX calculator or that an emulator is installed on your computer**
* that **you know how to use it** (if not : read **BINX** user tips manual to get and inkling or read the “quick start” official Hewlett-Packard user manual (download available on my web pages : just look attentively in Publication_3 page).
Reading the user’s tips of the other programs can do wonder to clarify some points about the use of the HP48 to run my programs, all of them follow an identical trail.
The learning curve of TOTU has been made quite gentle .

This program is easy to use even if it demands that you perform some work manually with paper and pencil. (**you need to create an array describing the layout of the crossings in the knot**)

With **TOTU**’s help you can **find the coding of each halt-period of any** (limit 32L / 32 B) **single strand knot following a turk’s head knot cordage route with whatever coding :**
--- **COLUMN** coded
--- **ROW** coded
--- Row **AND** Column coded
--- **NEITHER** Row **NOR** Column coded

For the first three types of coding the specialized programs dealing with those (already put online) are much faster and a lot more user friendly than this one (almost no paper and pencil work with them).

Still with the NEITHER row NOR Column coded type only this program can help you provided that that you have a diagram of the knot made on isometric grid paper in the style promoted by SCHAAKE and TURNER.

This program is my own mix things learned in **THE BRAIDER** and in
-- **BRAIDING THE REGULAR KNOTS** (this is the source of the diagram I use here In a modified form)
-- **THE BRAIDING OF LONG COLUMN CODED REGULAR KNOTS**
-- **THE BRAIDING OF ROW-CODED REGULAR KNOTS**

Now that the programming has been done any one of you can easily, with no hard mental work; adapt it to any other programming language of your choosing.

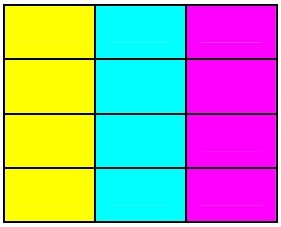
HOW DOES THIS PROGRAM WORK ?

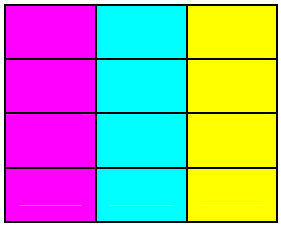
Simple enough :
YOU write the array describing the knot in a standardized and symbolic manner.
That array will be the variable named 'MALR'
This matrix describe in an isometric diagram the lay-out of the crossings in the knot diagram with their type (OVER UNDER) read as seen by an ODD numbered half-period going from low-left to up-right in the chosen frame of reference (horizontal mandrel. For the vertical cylinder the ODD numbered HP are running bottom-right to top-left)

This **MALR** array takes care of the ODD numbered half-periods reading the parallel bight line (column for mandrel, row for cylinder) from **LEFT** from **RIGHT**.

To provide for the EVEN numbered half-periods reading from **RIGHT** to **LEFT** the calculator will 'mirror ' the **MALR** matrix

Say this one becomes that one.





It should be obvious for every one that
--- this translate the **1…2…3……..n-1……n** that is at the top of the diagram which becomes at its bottom **n….n-1……3….2….1**

1	1	1
1	1	1
1	1	1
1	1	1

It remains to take on account that when an ODD numbered half-period read a given crossing as say OVER the EVEN half-period contributing in the making of this crossing read it as UNDER.
To take that in account the calculator will be built a "unit" matrix that will be subtracted from the mirrored MALR

9	0	9
1	9	1
9	0	9
1	9	1

MA**LR**

9	0	9
1	9	1
9	0	9
1	9	1

mirrored

8	-1	8
0	8	0
8	-1	8
0	8	0

After subtraction

8	1	8
0	8	0
8	1	8
0	8	0

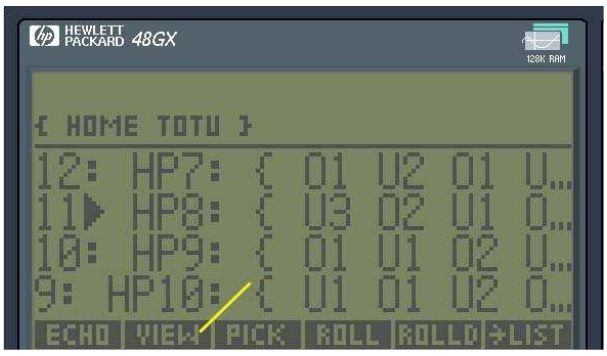
MARL

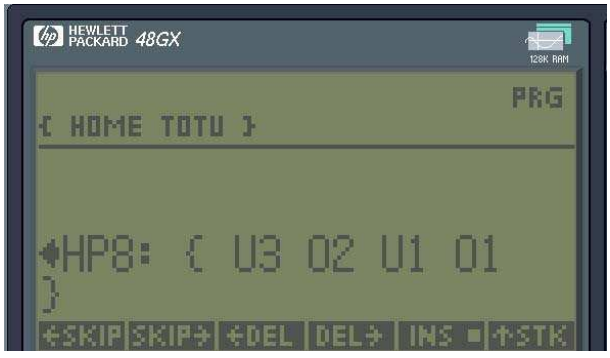
Now it just remain to take the ABSOLUTE of this post-substraction matrix to get the one that will be the MARL variable created by the calculator for the EVEN numbered half-periods.
The type of each crossing have been changed for its opposite.

Now the calculator will compute for each half-period the crossing it makes with the previously laid half-period and using it coordinate in COLUMN and ROW to get the type of the crossing will read the concerned matrix (**MALR** for the **ODD** numbered HP and **MARL** for the **EVEN** numbered HP).
As this is not rocket sciences and I don't envision any one having difficulties there.

Now say that at the end of **PGSU** one half period is written as
{ **0 0 1 0 1 1 1 0 1 1 0 0 0 0 1**}.
You may use it as it is the meaning being
{ **U U O U O O O U O O U U U U O**}

Running **TOTAL** (you will have to look for it in the menu at the bottom of the screen as the variables created by **PGSU** will have make the list longer) will rewrite it as
{ **U2 O1 U1 O3 U1 O2 U4 O1**} ; again not really rocket science.





Just make use of the ACTIVE **STACK feature**, select the HP you want and activate **VIEW** in the menu at the screen bottom and you will see this half-period in full.

What is rocket sciences is the calculation for getting the COLUMN and ROW of each crossing AS IT APPEAR in the laying of the successive HP : this is SCHAAKE & TURNER part !

In fact the rocket science part was discovering the “law” governing the crossing but the equations are not really hard (any 14 years old of my generation with the usual schooling would have tackled them without undue stress.
I cannot know what , say, an American or an English school boy can accomplish, but certainly it should be at least as good if not better.)

With the **TOTU** in your HP48 you can export it as ASCII and get to the source in clear TXT format (with some special HP characters mistakenly reproduced may be - make the correction by reading the ‘object’ in the EDIT of the HP48 (real or emulator).) If someone want it just email me.

RPL is a user friendly language and HP did wonder with that HP48.

HOW TO USE THE 'TOTU' HP48 PROGRAM

You loaded the **TOTU** program in either the calculator or the emulator and stored it .

TOTU is in the {**HOME**}, open the **TOTU** directory.

You will be using successively **TWO** programs **PGSU** and when PGSU is finished an optional **TOTAL** will be run.

I provided FOUR READY MADE ARRAYS for FOUR KNOTS : **MASEDG** , **MAT35**, **MAT37** and **MAT41.** I made those knots and tested those arrays and a lot more of them to verify the programs.

Before using any of the program (**and after** you are completely finished with a **TOTU** full run, so as to reclaim memory used by the many variables created) run the program **CLEAN.**

Just after CLEAN ran you need to STOre the array of the knot you want in 'MALR' (it is supposed that you know how to **RCL** and **STO** a variable using the **STACK)** :

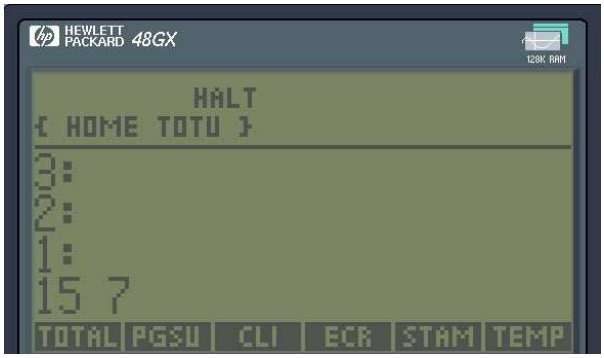
Let us say that for a first use you use **MASEDG** :

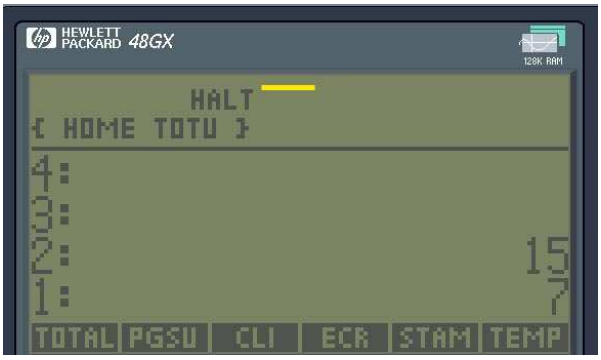
Put **MASEDG** on the stack by just pushing on the button just under **MASE** in the menu row

Type ' ' and inside write **MALR** (capitals) and **STO** it.

If MALR does not exist (or is not the correct one you want) you will get a program crash in the first case or results that are those of the array in the variable MALR but which is not the one you really want in the second case.

Now that you are sure that you have the **VALID** array in **MALR** variable run **PGSU**





You will be asked to enter LEAD and BIGHT numbers for this knot.

Type the numbers separated by a space then push the ENTER button
This **HALT** the program.

To resume the running of the program while use **Left-side violet arrow button** + **ON**.

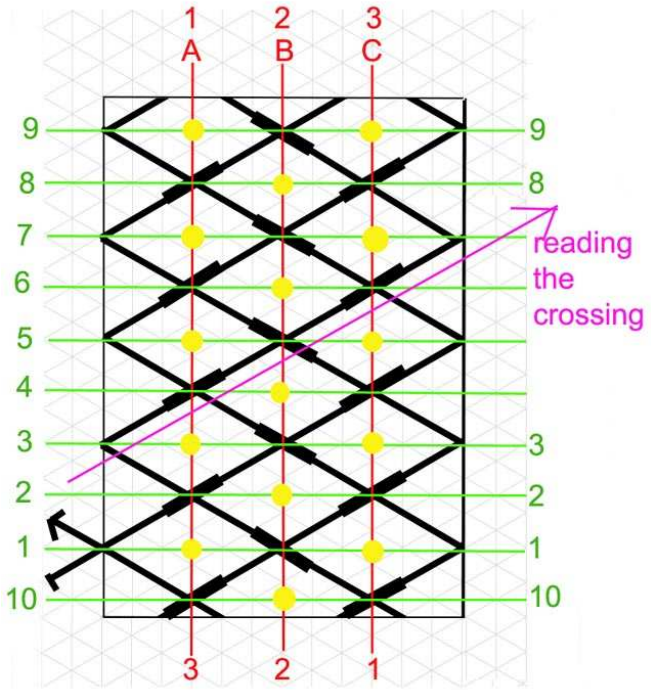
An hourglass icon appears in the top right side of the screen meaning calculation are Being done.

Wait till the hourglass on the upper part of the screen disappear. (very fast, may be 10 seconds, with the emulator, slow with the calculator : can take 1 to 10 minutes depending on the size of the array.

Results will be put in the **STACK** where you can access them with ease.

FIRST EASY LESSON ON WRITING THE KNOT's ARRAY

I made an especially easy diagram: **4L 5B regular THK**



MANDREL FRAME of REFERENCE ON AN ISOMETRIC GRID

You MUST make your diagram on an ISOMETRIC GRID to easily see the alignments crossing.

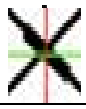
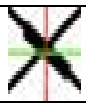
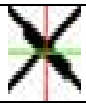
	COL #1	COL #2	COL #3
	COL A	COL B	COL C
Row #1			
Row #2			
Row #3			
Row #4			
Row #5			
Row #6			
Row #7			
Row #8			
Row #9			
Row #10			

Please note that the blank white coloured cells in the table are in the same number as are the intersections between the **RED** lines and the **GREEN** lines in the diagram.

The numbering in **RED** that will be use to write the table are the digits at the top (they are also letters for the sake of illustration but array/matrix have columns and row identified by digit and not number in the HP48)
They are numbered 1 to 3 that is (**L – 1**)

Note also that the horizontal **GREEN** lines are numbered 1 to 10 (10 == (**2 * B**))

Now we enter the intersection between **RED** and **GREEN** lines that show no knot crossing and are denoted by a YELLOW circle in the isometric diagram. In the array they will be '**9**', the OVER crossing being '**1**' and the UNDER crossing being '**0**'

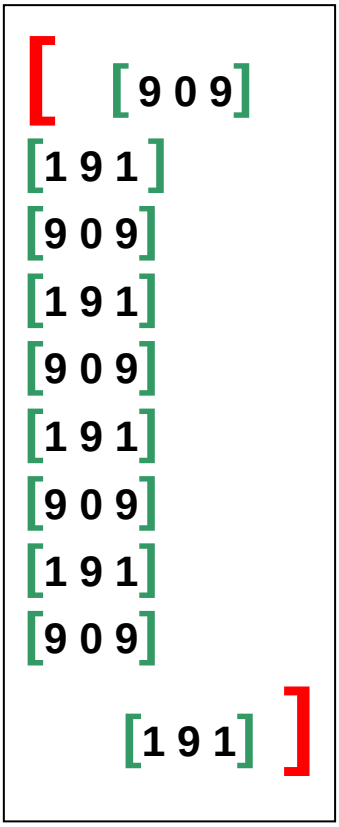
	COL #1	COL #2	COL #3
	COL A	COL B	COL C
Row #1	9		9
Row #2		9	
Row #3	9		9
Row #4		9	
Row #5	9		9
Row #6		9	
Row #7	9		9
Row #8		9	
Row #9	9		9
Row #10		9	

	COL #1	COL #2	COL #3
	COL A	COL B	COL C
Row #1	9	0	9
Row #2	1	9	1
Row #3	9	0	9
Row #4	1	9	1
Row #5	9	0	9
Row #6	1	9	1
Row #7	9	0	9
Row #8	1	9	1
Row #9	9	0	9
Row #10	1	9	1

We get that simplified table_____
That you now have to write in the HP48 in a matrix form

9	0	9
1	9	1
9	0	9
1	9	1
9	0	9
1	9	1
9	0	9
1	9	1
9	0	9
1	9	1

[[Row 1] [Row 2] [Row 3] [Row n-1] [Row n]]



It is easy to make a mistake when entering an array.

Enter it TWICE
(each time STOre it under a different variable name).

That done ReCaLl each of the array on the STACK then type SAME, push ENTER.

If the two arrays are identical then you get ' **1** ' as answer or ' **0** ' if they are different.

If they are different AT LEAST ONE is mistaken.

If they are identical then they are either both correct or both mistaken: just verify attentively one of them to see if it is exactly as you written table.

Failing to do that validation (of the diagram, of the writing of the array on the paper, of the writing of the array in the calculator, of the concordance between diagram and array entered) **may lead you to get mistaken half-period coding : calculation will have been correctly made on the wrong material !**

Now for some "real life" diagrams and arrays.

REMINDER

In *THIS* frame of reference (horizontal mandrel) the BIGHT rim are on the **LEFT** and on the **RIGHT** (*not* TOP & BOTTOM) side of the isometric diagram

The **SP**art / **WE**nd crossing is on the **LEFT** side

Your paper and pencil table will need (**L- 1**) columns and (**2 * B**) rows

"absence " of crossing at an intersection of a vertical line with an horizontal line of crossing is denoted ' **9** '

An OVER crossing (as noted by an EVEN numbered half-period) is denoted ' **1** ' and an UNDER crossing by ' **0** '.

HOW TO WRITE THE FOUR ARRAYS PROVIDED FROM THE KNOT DIAGRAM

It is for this type of **NEITHER ROW NOR COLUMN CODED** knot that this program is the most useful.

MASEDG

```
[[ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 0 9 1 9 ]]
```

ARRAY will have

(L-1) Columns numbered left to right from 1

(2*B) Rows numbered top to bottom from 1

'0' stands for an UNDER crossing.

'1' stands for an OVER crossing.

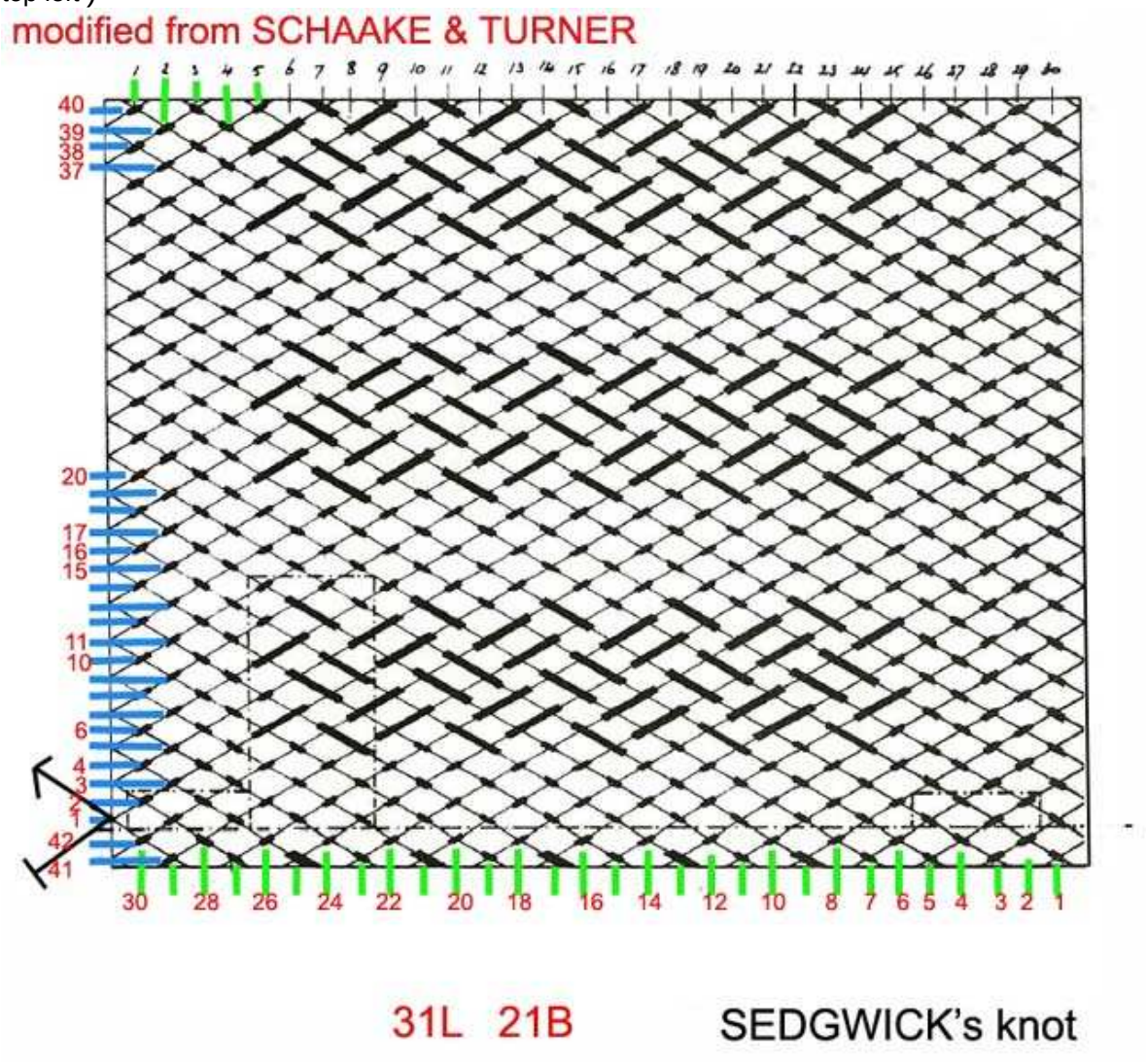
'9' stand for 'empty space' at this particular place of the row or the column in the knot diagram.

NOTE = the type of the crossing, OVER or UNDER, is read as the ODD numbered half-period going from low left to up right would read it.

Train yourself with the four examples given.

Now for the only tricky point : Numbering the columns and rows on the knot diagram (we use here the horizontal mandrel frame of reference

with the bight rim on the left and on the right .(vertical cylinder would have the bight rim top and bottom and the half-period reading the type of the crossing would be going from bottom right to top left)



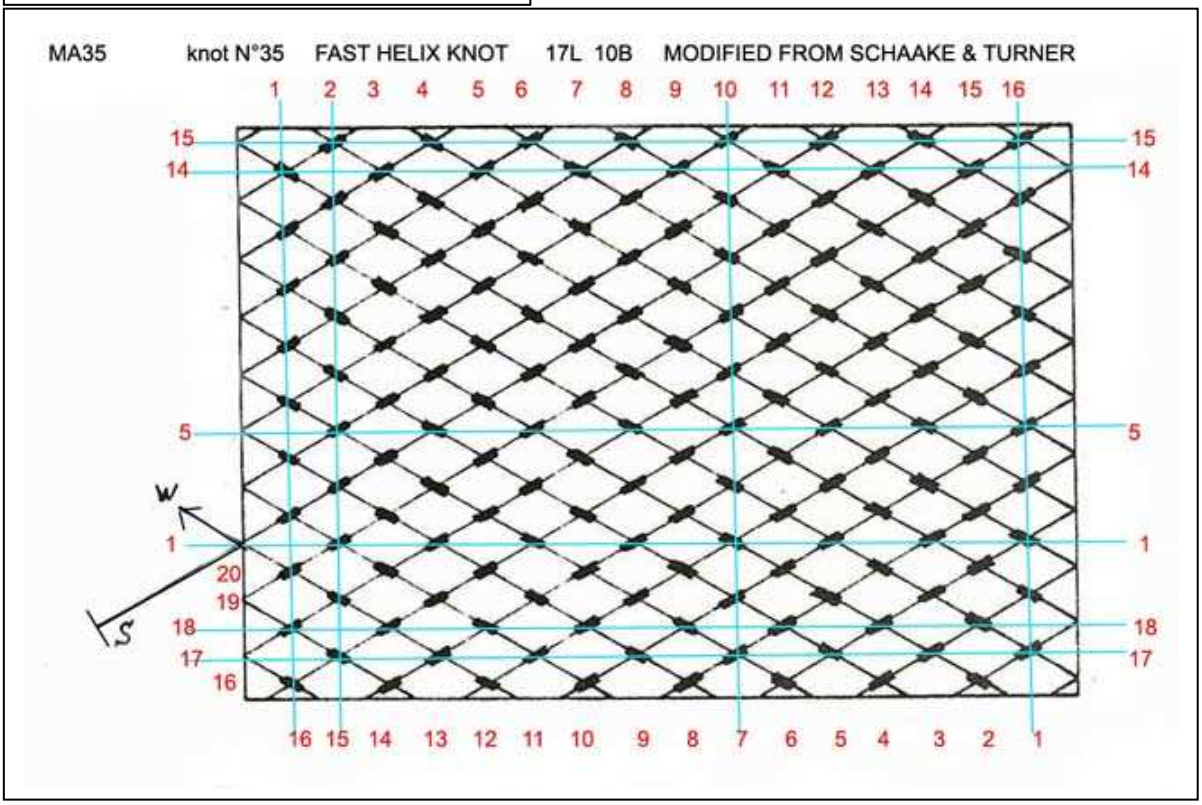
For the ARRAY to be entered in the 'MALR' variable you will number its columns as the columns of crossing are numbered from left to right at the top of the diagram 1, 2, 3 to L-1

Now the rows of the ARRAY will be numbered from 1, 2 , 3 to (2*B) **BUT how do we find the correct numbering ON THE KNOT DIAGRAM ?**

Using the horizontal mandrel reference and remembering that the bight rim of a real knot is a circle, in the diagram, we put **ROW N°1 where the Wend cross the SPart and go on upward**.
When at the top ROW then we go down to the lowest ROW and continue the numbering.
Just study the diagram.
NOTE : this numbering I use here has nothing to do with all the numbers SCHAAKE system use. I made it such so as to be able to write a valid array / matrix for the HP (in an array numbering or row and column does not begin at '0' but at '1'. There is no use of modulus here that would make mandatory a numbering beginning at zero.

MAT35

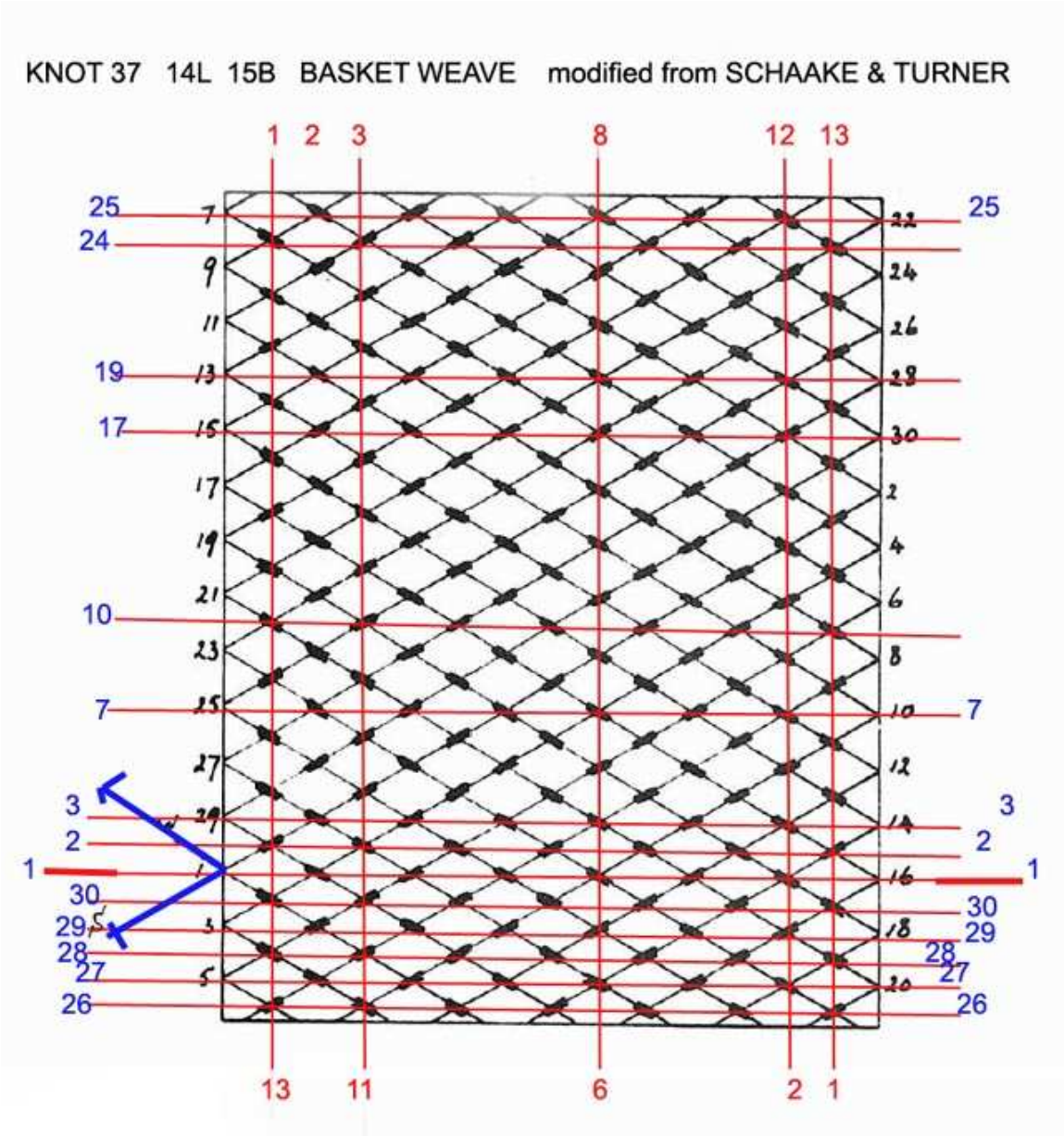
```
[[ 9 1 9 1 9 0 9 1 9 0 9 1 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 0 9 1 ]
 [ 0 9 1 9 1 9 0 9 1 9 0 9 1 9 1 9 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 1 9 0 9 1 ]
 [ 0 9 1 9 0 9 1 9 1 9 0 9 1 9 0 9 ]
 [ 9 0 9 1 9 1 9 0 9 1 9 0 9 1 9 1 ]
 [ 1 9 1 9 0 9 1 9 0 9 1 9 1 9 0 9 ]
 [ 9 0 9 1 9 0 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 9 1 9 1 9 0 9 1 9 0 9 1 9 1 9 0 ]
 [ 1 9 0 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 0 9 1 ]
 [ 0 9 1 9 1 9 0 9 1 9 0 9 1 9 1 9 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 1 9 0 9 1 ]
 [ 0 9 1 9 0 9 1 9 1 9 0 9 1 9 0 9 ]
 [ 9 0 9 1 9 1 9 0 9 1 9 0 9 1 9 1 ]
 [ 1 9 1 9 0 9 1 9 0 9 1 9 1 9 0 9 ]
 [ 9 0 9 1 9 0 9 1 9 1 9 0 9 1 9 0 ]
 [ 1 9 0 9 1 9 1 9 0 9 1 9 0 9 1 9 ]].
```



MAT37

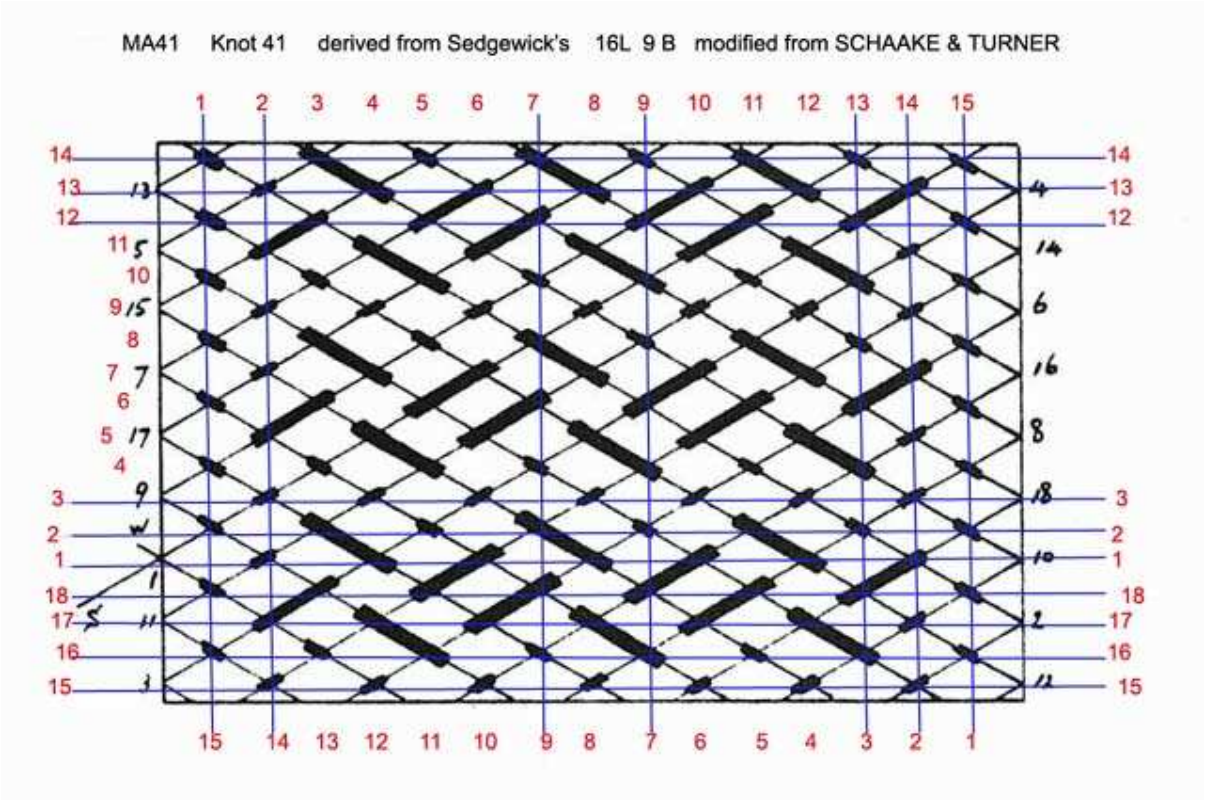
```
[[ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 1 9 0 9 0 9 1 9 0 9 0 9 1 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 1 9 0 9 0 9 1 9 0 9 0 9 1 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 1 9 0 9 0 9 1 9 0 9 0 9 1 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 1 9 0 9 0 9 1 9 0 9 0 9 1 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 1 9 0 9 0 9 1 9 0 9 0 9 1 ]
 [ 9 0 9 1 9 0 9 0 9 1 9 0 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 0 9 1 9 1 9 0 ]]
```

9	0	9	1	9	0	9	0	9	1	9	0	9
1	9	0	9	0	9	1	9	0	9	0	9	1
9	0	9	1	9	0	9	0	9	1	9	0	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	1	9	0	9	1	9	1	9	0	9	1	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	0	9	1	9	0	9	0	9	1	9	0	9
1	9	0	9	0	9	1	9	0	9	0	9	1
9	0	9	1	9	0	9	0	9	1	9	0	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	1	9	0	9	1	9	1	9	0	9	1	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	0	9	1	9	0	9	0	9	1	9	0	9
1	9	0	9	0	9	1	9	0	9	0	9	1
9	0	9	1	9	0	9	0	9	1	9	0	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	1	9	0	9	1	9	1	9	0	9	1	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	0	9	1	9	0	9	0	9	1	9	0	9
1	9	0	9	0	9	1	9	0	9	0	9	1
9	0	9	1	9	0	9	0	9	1	9	0	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	1	9	0	9	1	9	1	9	0	9	1	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	0	9	1	9	0	9	0	9	1	9	0	9
1	9	0	9	0	9	1	9	0	9	0	9	1
9	0	9	1	9	0	9	0	9	1	9	0	9
0	9	1	9	1	9	0	9	1	9	1	9	0
9	1	9	0	9	1	9	1	9	0	9	1	9
0	9	1	9	1	9	0	9	1	9	1	9	0


KNOT 37 14L 15B BASKET WEAVE modified from SCHAAKE & TURNER
1 2 3 8 12 13
25 24 19 17 10 7 3 2 1 30 29 28 27 26
25 7 9 11 13 15 17 19 21 23 25 27 29 1 3 5
22 24 26 28 30 2 4 6 8 10 12 14 16 18 20
25 7 3 2 1 30 29 28 27 26
13 11 6 2 1

MA41

```
[[ 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 ]
 [ 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 ]
 [ 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 1 9 1 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 ]
 [ 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 ]
 [ 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 1 9 1 9 1 9 1 9 0 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 ]
 [ 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 ]
 [ 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 ]
 [ 9 1 9 0 9 1 9 0 9 1 9 0 9 1 9 ]
 [ 0 9 1 9 1 9 1 9 1 9 1 9 1 9 0 ]]
```



Here is this Knot 41 (a derivation from the Sedgwick's)