BINX

or YAJOAT yet another JOAT

BUT MUCH MORE INTERESTING

The computation procedures are different from those used in JOAT and rest on Schaake & Turner's work once again.

Bight Index Numbers (what I had in my own quest labelled otherwise in my THK pages).

The computation procedures being different it will allow the untrusting souls to verify the JOAT results and even those of the other programs results.

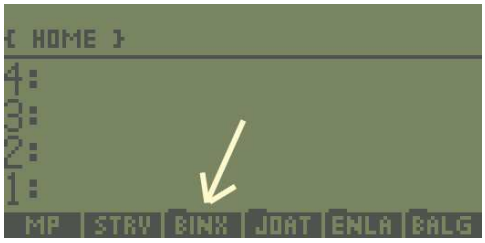
It is till EMU48 for Windows and HP-RPL language.

This one **will do ROW-coded, COLUMN-coded, ROW&COL-coded, NEITHER ROW NOR COL-coded.**

READ JOAT USER'S TIPS AGAIN PLEASE you will need that to understand without trouble what is following and to "plug" the holes I have left in my explanations.

FIRST OF ALL READ THE ANNEXE AT THE END OF THIS PAPER BEFORE TACKLING THE MAIN TEXT.

We are in EMU48 with all the programs "loaded".

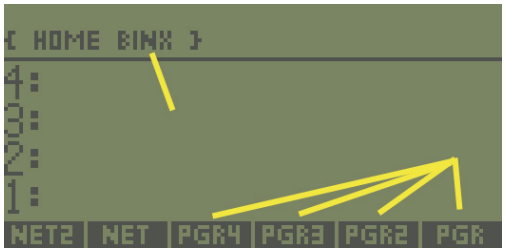


RESULTS WILL NOT BE CODING SEQUENCES OF CROSSINGS BUT IDENTIFYING NUMBER FOR COLUMNS OR/AND ROWS WHERE CROSSINGS HAPPEN.

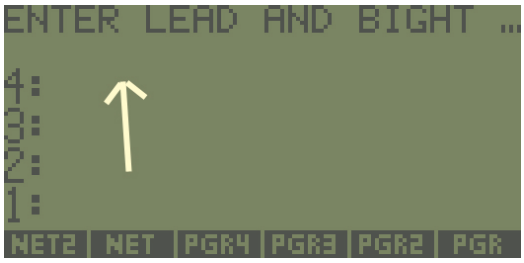
Among the {HOME} DIRectories you will **find BINX's**.
If you cannot see it then use the side arrows to try and find it.

Click on it to open it.

Now you are inside BINX DIRECTORY as



can be seen in the uppermost line on the screen: { HOME BINX }.
You can see (if not use side arrows) the four "staged" programs :
PGR, **PGR2**, **PGR3**, **PGR4**
(a week of pondering and researching went into the naming so don't snort please). **PGR is indeed PGR1.**
So you "push" on it to **RUN the program**.



PGR(1) immediately retaliates for having been aroused from a pleasant slumber and asks, no demands that you **enter number of LEAD and number of BIGHT** and makes a HALT (the uppermost part of the screen).

Do enter what has been ask from you.

You have to **KILL the HALT** (equivalent to FORWARD MARCH, MARCH) **using the violet Left arrow and the ON keys.**

Program runs and puts the **results** of its calculations **in the STACK**.

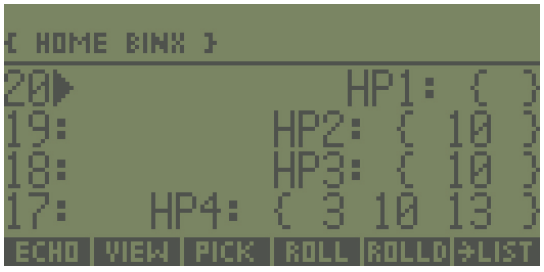
The results are expressed under the form of COMPLEX numbers : have no fear they do not bite and anyway I am using them here as “label” and not as complex numbers.

You get **a whole collection of (SOME_NUMBER_1 , SOME_NUMBER_2)**
SOME_NUMBER_1 stands for BIGHT-INDEX
I_0, I_1, I_2 … I_B (whether left or right – mandrel frame of reference-)
ODD are on the LEFT BIGHT RIM (**left** to **right** H-P) and **EVEN** are on the RIGHT Bight rim (**right** to **left** H-P).

Above “21” flags one that is on the **LEFT** BIGHT RIM
and
“20” goes with one that is on the **RIGHT** BIGHT RIM.
SOME_NUMBER_2 is the computed Bight-Index-Number.

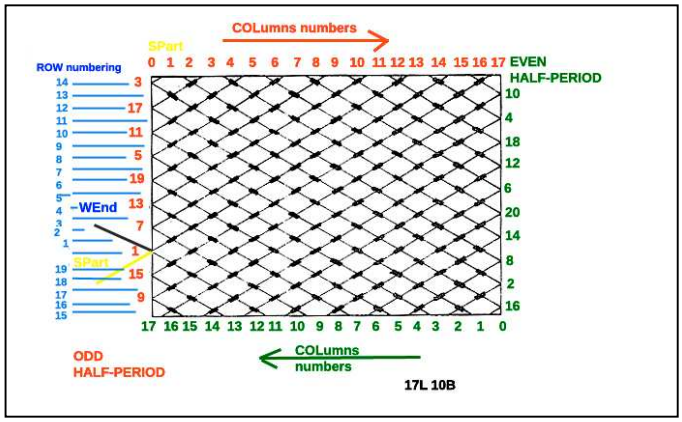
Now if you don’t want to stop here and ‘manually’ exploit the results obtained you can run **PGR2**.

RUNning PGR2 will give you **for EACH**

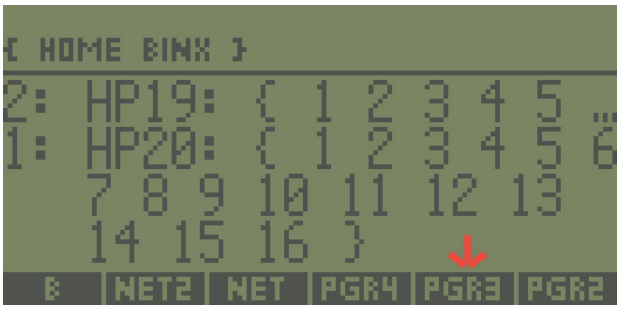


H-P in the knot the number of the COLumns where a crossing exist for that Half-Period.
HP4: { 3 10 13 } means
Half-Period N°4 has a crossing in COLumns N°3, N°10 and N° 13.

You may stop here and make use of those result on your “prepared diagram”



You may also decide to push you luck with the HP48GX and run PGR3.



PGR3 will gives you a list of numbers (empty { } lists are for FREE RUN remember ?)
that are labelled **CROW** (for Crossing ROW)

For EACH H-P just as before you got the number identifying the columns where a crossing existed , here **you get the**

identifying number of the ROWs where a crossing exists.

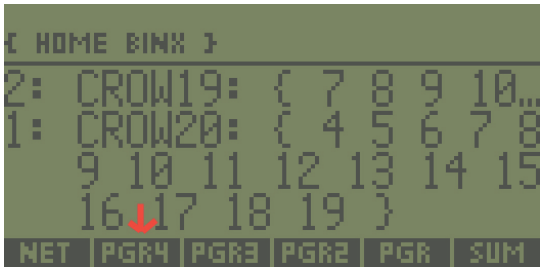
CROW4 is 4th H-P, CROW7 is 7th H-P. CROW4 or 4th Half-period has a crossing in ROW N° 14, in ROW N° 1 and in ROW N°4. (**DO NOT** try to make that an 'ordered list' for ease of use as in { 1 4 14 } instead of using what directly resulted from the computation loops.)

Now there only remain to "mix" but in a carefylly appropriate way the N° identifying COL and the N° identifying the ROW to get the intersection where the crossing is : Now it will only be necessary to read the TYPE (HIGH / LOW over/under) of this crossing on the diagram prepared prior to using the programs.

Please remember that the "perspective" on the TYPE of a crossing depends on the way the arrow SPart-Wend is going : **LEFT** to **RIGHT** for an **ODD** Half-Period, **RIGHT** to **LEFT** for an **EVEN** H-P :

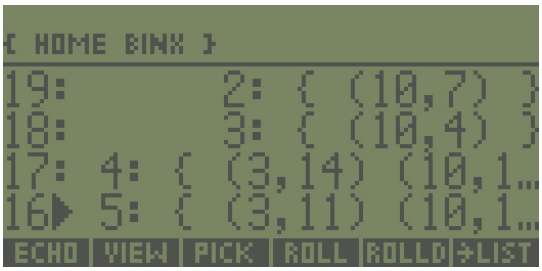
the OVER in the perspective of one will be the UNDER in the perspective of the other.

Better not forget that .(even when writing a program : that is why it is much better to use 0 and 1 that can be easily convert to the opposite just using a 2 MODULO than persisting in the very cumbersome / & \.)

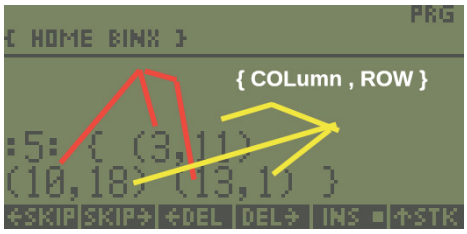


RUN **PGR4** to get those intersection coordinates per H-P.

PGR4 gives for each crossing a set of number identifying COLUMN and ROW where a crossing happens for the H-P studied.



Again COMPLEX numbers (but for you they will just be labels)



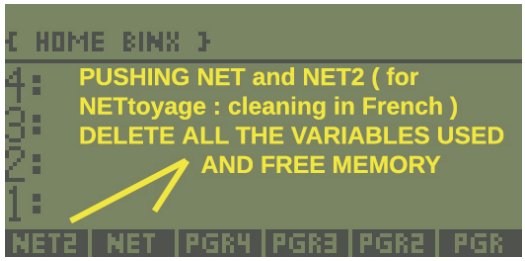
Those are in fact COORDINATES for EACH CROSSING in a given HP (Half-Period) first you get the COLumns (remember read LEFT to RIGHT for ODD HP running from left bight rim to right bight rim and RIGHT to LEFT for EVEN period running from right bight rim to left bight rim.

Second number identify the ROW (again a reading order has to be complied to : ROW **ZERO** is at the **SPart** in the knot or the return of the **WEnd** to the **SPart**, the numbering is made upward and "in a circular fashion" : remember always think MODULO : look at the row numbering in the prepared diagram)

Use that manually to get the **CODE SEQUENCE OF CROSSING** for each HP in the knot you have the diagram of.

Of course a PGR5 asking for extensive entry of code per row or per columns or for the whole knot can be written that will write automatically the coding sequence of crossings (may be I will do it but for NEITHER / NOR coded it will need very heavy entering of data by user)

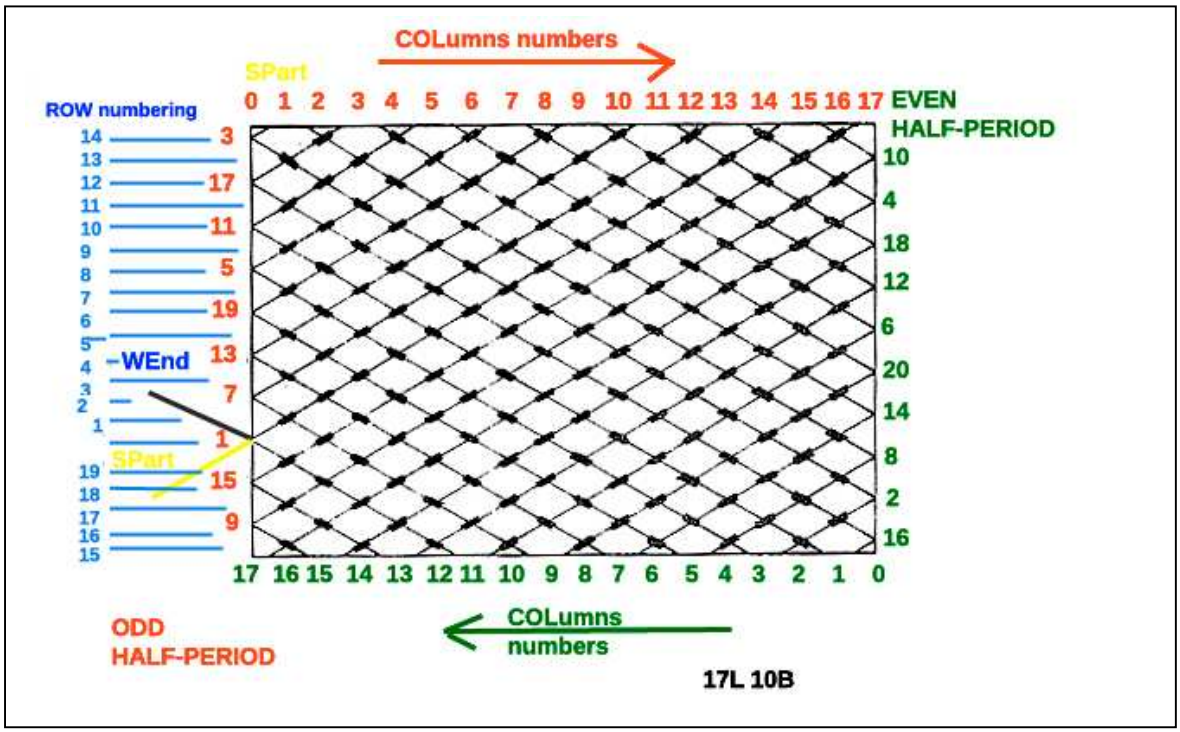
Now that you are finished with playing : it is time to CLEAN the mess you made !



NET (abbrev NETtoyage ==cleaning in French) and NET2 (in whatever order takes your fancy) will free the memory (not important on the computer but really important on a real HP48GX calculator). NET and NET2 are RUN when the programs RUN if only to avoid bugs due to pre-existing variables

---

ANNEXE



Instead of putting ALL the H-P in the same layer as in a classical diagram (see above) they are “unrolled” as I explained in :

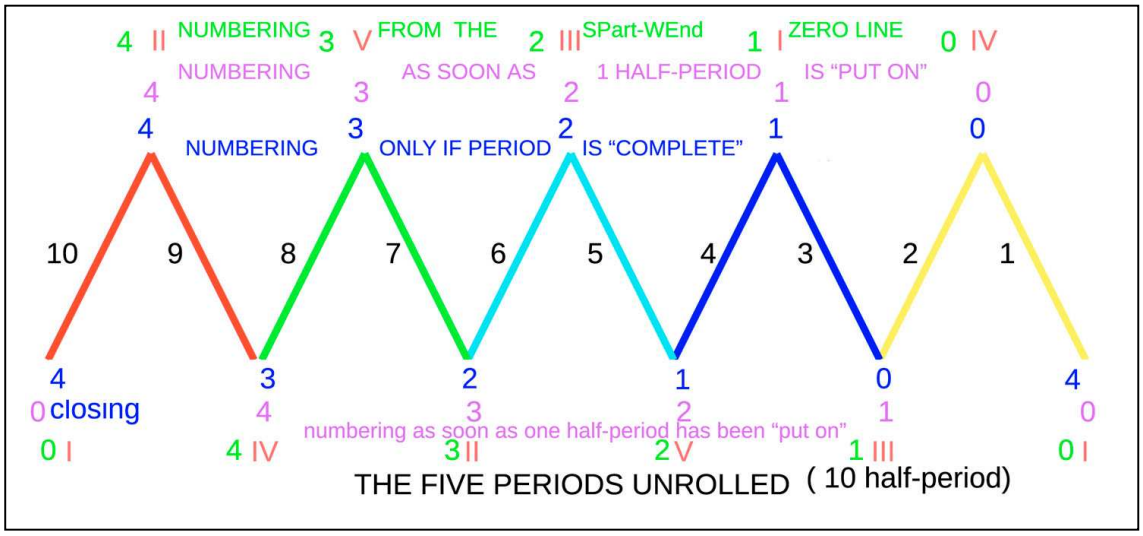
--- http://charles.hamel.free.fr/knots-and-cordages/turksheads_3.html
as it relate to ENLARGEMENT in topic :
UNROLLING OF TIME SEQUENCE COMPARED WITH
SPATIAL SEQUENCE FOR ENLARGEMENT PROCESSES

--- in http://charles.hamel.free.fr/knots-and-cordages/turksheads_4.html
in
BIGHT NUMBER versus BIGHT ORDER

--- http://charles.hamel.free.fr/knots-and-cordages/turksheads_5.html
in particular in
HOW TO GET THE ORDER IN WHICH THE BIGHT ARE COMPLETED
IN THE PROCESS OF MAKING A TRUE THK:

--- http://charles.hamel.free.fr/knots-and-cordages/turksheads_6.html

HOW TO COMPUTE THE ORDER IN WHICH BIGHT ARE IN FACT LAID BY THE RUNNING CORDAGE



THEIR TIME SEQUENCE IS UNROLLED IN SPACE and a certain computation is done.

Now for SCHAAKE & TURNER perspective on that.