

CG Tabs Updates Version 3.7

Attached are some brief additions to the previous documentation from the previous version:

Hide/Show Judges' Ranks:

This button will temporarily hide the judges ranks (by making cell contents white) to avoid having private ranks displayed accidentally. The sorted column is also deleted. Showing ranks makes cells black and resorts. There is a line that indicates whether judges ranks are displayed or hidden - do NOT delete this line!

Sort/Scramble Round:

This new macro will automatically put the round sheet in room rank order to allow for easier manipulation of judging assignments. If you scramble them again, the tab will be put back into the SAME ORDER it was in before you sorted it. There is a line that says whether the tab is ranked or scrambled - do NOT delete this line!

Resolve Judging Conflicts:

All judging conflicts are automatically resolved when judges are added, or this macro can be run separately if needed also. After judges are assigned, and a conflict is detected, the list of judges will be searched to find a judge, WITH AS CLOSE OF A RANK AS POSSIBLE, to swap with. Should you create a conflict by manually moving judges, you can run the macro to automatically remove it. This works no matter what the method of judges assignments was and will maintain the same number of judges per rooms. Conflicts are determined by the conflict matrix which you enter manually.

New Sorting Tab Features:

First of, ranks are now automatically added when you use this macro. To avoid issues of scrambling teams that make the break with those that don't, there is a new column called "Break" at the end of the "tabsheet". If teams make quarters, then enter "q" in the cells for the 8 Quarter-Finalists teams. If you sort the tab all teams with this "q" will rank as the top 8 teams, and then the rank will be "1-Quarters", etc. Then, if desired, for any teams who make semis, put an "s", although this is optional. A finalist is denoted with "f" and tournament champions as "c". This will make sorting easy! Also has the option for "o" for octo-finals. So, in the sort tab, "c" will always be the top rank, followed by "f", "s", "q" and "o" in that order. For all others leave it as "n" for did not break. This is totally optional but does solve a problem that people encountered by forgetting to manually move teams in the break.

New Judging Assignments Type: (very slick idea)

This is entered as judging type 3 and was designed in large part by Vinay Mysore. It works best for tournaments or rounds that must have odd numbers of judges, and/or longer tournaments in which more sophisticated methods of slotting judges are needed for later rounds. For the random round, which can be round 1 only, it will put an odd number of judges only while spreading out the top judges as much as possible in a random order. If there is an extra judge then there will be 1 room with an even number.

For bracketed rounds, what this is effectively doing is combining the two previously designed methods for bracketed rounds. When judges are "stacked", you are putting the top three judges in the top room for example, the next best 3 in room 2, etc. When you "spread" them out, it will take your top 12 judges for example and make them the chairs in all 12 rooms in order, than fill up with the next best judges as the second chair, etc. "Vinay's" Method allow you to specify a very specific number of rooms which are important, and put your top judges in those rooms. It effectively breaks the tab in two, and then allows you to put different numbers of judges per room, always keeping an odd number. You must, however, know exactly how many judges you have, how many rooms are in the "top" and the "bottom", which method to use, etc. From a practical point of view, you would probably probably spread out in the top rooms, but could use either method for the bottom rooms. The following is an illustration of the data that must be entered:

CUSID Style with Odd Judges: (Vinay's Method)					
Verify Accuracy of Entries					
	Method	Start	Finish	Max. Jud.	Total Jud.
Top Rooms:	1	1	6	5	30
Bottom Rooms:	2	7	11	5	25
Automatically resolve conflicts?			1		
Method 1 is spread out, Method 2 is stacked.					

What will happen in this example is that the top 30 judges will be put 5 per room in the top 6 rooms. The best 6 judges will chair, the next best six will second chair, etc. In the bottom 5 rooms, the next 25 judges will preside with the best 5 of that lot “stacked” into the number 7 room, on down to the bottom 5 judges in the bottom room. In practice you may want to spread out judges for the bottom rooms. Always remember that judges with the same rank are sorted randomly.

Also, there need not be an equal number of judges in each room in the top half, or bottom half. If I entered 5 as the max judges, but then only 28 judges, then the top 5 rooms would have 5 judges, but the number 6 room would have 3 judges. If I entered 27 or 29 judges, this will not work as an odd number per room is not guaranteed. You could enter as low as 17 judges. This will put 5 in the number 1 room, but only 3 in the other rooms. If you had only 15 judges for the top, then you would need to change max judges to 3. The program checks number very carefully before proceeding, and the “Verify Accuracy” will tell you if data is right before proceeding. If not, your specific error(s) is identified. You may ONLY put max judges at 1, 3 or 5. Whichever you pick, all rooms in that block will either have max judges, or that number minus two to ensure an odd number. For the bottom rooms, you once again must check the number of rooms and the max judges and then ensure total judges make sense. However, there need not be a number of judges that ensures an odd number - if there is an (one) extra judge he/she will be slotted and one room will have an even number.

Many cool things can be done with this. You could for example slot your top 12 judges in hidden quarter-finals with only 1 judge per room after that. You can use the “spread” method for both the top and bottom. You could for example specify 5 judges per room for the top rooms, and then 3 per room for the second block(and depending on numbers if may fill out bottom rooms with singles.) If you want to manually resolve conflicts you can disable that feature by putting “0” in the box. However, the algorithm is quite good, although you can never be 100% where the conflicted will move to.

What I did at our provincials was the following. I knew we had 11 rooms, and 27 judges for the final rounds, which meant 8 rooms with 3 per room, and then 3 singles. I thought a bit about which I wanted as singles and decided to do the following. I selected the top SIX rooms to have 14 judges. I knew that this would produce 4 rooms with 3 judges, and then 2 rooms with 1 judge, and knew that these rooms would be the 5th and 6th ranked rooms using the 5th and 6th ranked judges. I did this to give very good judges to reasonably important rooms, but knowing they couldn't win. I then left 13 judges for the bottom 5 rooms, and used the spread out method as well. I knew this would put one judge only in the bottom, but it would not be weakest judge overall. The assignments were done pretty much perfectly, with some conflicts to resolve.

As you can see it is totally flexible, although perhaps somewhat complicated until you actually use it. Hopefully the example illustrate what is possible.