**Introduction**

This is my Christmas Dinner time plan spreadsheet, that I have been using for several years. I was asked if I would share it, and am happy to do so. But I do so on the understanding that I am unable to provide any warranty or support whatsoever. If you wish to provide feedback, then feel free to do so. If there are issues that seriously affect functionality, then I will try to address them, but cannot promise to. If there are requests or suggestions for improvements or further functionality, I might consider them if I have time. You are free to amend and use in any way you see fit, but not to sell or redistribute without permission of the author/developer. If you do make changes that you think would be of benefit to others, then please consider publishing them in the GitHub repository, or at least email them back to me.

© Lee Hill, 2023. [XmasTimePlan@leewhill.com](mailto:XmasTimePlan@leewhill.com).

**Usage**

Each row in the spreadsheet represents an action that needs to be carried out at a particular point in time. The rows are grouped using the ‘item’ column, for which I use the name of the food item being cooked, e.g. carrots. In the location column I enter the ‘location’ where the item is being cooked, e.g. oven 1, or hob. The +/- column indicates whether this item is being put in/on to that location (+) or removed (-). If an item involves a “change of location” (e.g. remove potatoes from hob and place tray in oven) I split it into two rows, one with a “hob -“ and one with an “oven +” (for example).

In the ‘minus’ column, the user should enter the amount of time BEFORE the serving time that each action is to be carried out. At cell D4 you should enter the proposed serving time, and the formulas in the Time column will then work out the exact time at which the action needs to occur. Sorting the completed list on this column gives you your time plan.

The spreadsheet contains a timer, and a formula in the Due will change the value of this cell to ‘y’ once each item becomes due. This will change the formatting of the entire row, so that you can easily see which items are due. The exact formatting can be changed on the Parameters sheet.

The formulas will not recalculate dynamically, so once you start cooking, you should click the Start Monitor link, to set the timer running. This refreshes the page according to the frequency that you enter on the Parameters sheet.

When setting up your plan, it makes sense to go item by item, and enter a row for everything that needs to be done for that item. If you filter on the item ‘carrots’ in my list, you’ll see what I mean.

There is no reason why this tool can only be used for Christmas dinner; it can of course be used for any complex meal.

**Filters**

You can filter by each location in turn to see everything that is using that location, to check whether you are going to have enough room. This also helps to check you have the right oven temperature for each item.

You can also filter on the item column to see everything that needs to done for that item.

**Restrictions**

I have not yet added any cell-level protection, so it is currently possible to overtype all of the formulas that are needed for proper function. For this reason, it is important that you only enter data in cells A, B, C, E, F and G. Do not amend D or I, since they contain formulas. Column J is used during cooking; enter a y (case insensitive) when item is complete. Columns to the right of column J also have formulas, so don’t amend these.

I have left my own plan in to help you understand how it works, but once you do, you will want to delete all my data. Please ensure you do not delete the contents of columns D, I or M, or L to P. I have added a Clear function (see link in cell H3) which will get rid of all data except for the first row. Use carefully! You can then overwrite the detail of the first row with your own data.

Columns L to Q are an attempt to calculate how many items are in each location at any one time. They might be useful to you.

**‘Dynamic’ items**

There are two places at the top to specify details of two different meats. These of course do not have to be meats; they could be any item, or they could be left unused.

I enter the names of two main items in cells C2 and C3, and calculate the roasting and resting times, then enter those at D2 to E3. After that, in the item column for each row for these items, I enter the formula =Meat1 or =Meat2. In the Minus column, I can then use the variables Rest1, Rest2, Roast1 and Roast2. So Meat1 needs to come out of the oven at =Rest1. It needs to go into the oven at =Roast1+Rest1. The oven needs to go on at =Roast1+Rest1+”00:20” (allowing 20 minutes for the oven to preheat).

The reason for this functionality is that the timings for meat are dependent on weight, and very often the exact weight of the meat is not known until a day or two before Christmas. This allows the weight to be ‘fine-tuned’ at the last minute without having to amend each row. Remember to re-sort the list after changing any of these values.

Using this system requires a little more experience of using Excel. Hopefully you can work it out from my examples. If not, abandon the named range references and just use the item and minus columns for meat the way you do for everything else.

**Changes**

I have updated this to be as accommodating of any changes you want to make as possible, but without any cell protection, there are always things that a user can do that might break the code/formulas. If you just use the existing rows and columns, you should have no difficulties. You *should* be able to insert columns without any problems, but be aware this *might* break the functionality.

Adding rows is more problematic. The functionality relies on formulas, some of which are hidden. If you insert rows either in the middle or at the bottom, the formulas are not automatically added to the new rows. To combat this, I have added a Reformat routine, which runs when you click the link in cell H2. It is useful to run this after inserting or deleting rows, to ensure that everything is still formatting correctly. This routine attempts to ensure all the formulas and formatting are correct, but it is not foolproof! I can’t offer any kind of warranty or support for this, so the best thing to do is to save the original version as a master copy, from which to restore if you have problems, and then turn AutoSave off, but save frequently as you go.

**Code**

The VBA modules are currently uncommented. I am well aware that this is terribly bad form, but I don’t have time right now. They aren’t massively tricky, but there is one place in particular (where the formulas are inserted into row 9) that is a bit opaque