Discussion-1 Flowchart and Algorithm

Discussion Topic:

Describe an algorithm for baking cookies. Assume that you have many hungry friends, so you'll want to produce several batches of cookies! Include a brief flowchart for your algorithm as well. In response to your peers, provide at least one adjustment to make to their provided algorithm or flowchart.

My Post:

Hello Class,

Designing an algorithm for baking cookies turned out to be a bit more difficult than I thought, especially when you add "several batches of cookies!" Additionally, I opted for a detailed logical representation of the process, instead of a high-level representation of it, making this exercise a bit more challenging than I wanted it to be!
Anyhow, my baking cookies algorithm, below, is a detailed description of the different Phases involved in the baking cookies process.

0- Start

1- The '`Initializes Variables`' phase initializes several variables to zero. These variables are used as counters and to store different calculation results. The variables are:

- `total_cookies_needed`: stores the total cookies needed to feed all the friends
- `total_cookies`: As we are using batches of cookies that follow a recipe that uses a set of ingredient values proportional to each other, the number of cookies actually produced by those batches can be ≥ `total_cookies_needed`, and it should not be > `total_cookies_needed`
- `total_batches`: The total of mixed dough batches to produce total_cookies
- `trays_total`: The total number of cookie trays needed to accommodate `total_batches`
- `num_oven_runs_needed`: The number of cycles of oven runs necessary to accommodate `trays_total`
- `trays_in_oven`: The number of cookie trays being baked in the oven, which can be ≤ `oven_slots`, and it cannot be > `oven_slots`
- `mixed_dough`: The amount of mixed dough to produce `total_cookies`
- `i`: counter used to keep track of the number of batches created
- `n`: index counter used to keep track of ingredient index in the `ingredients_per_batch` list

2- The '`Inputs Capture`' phase captures user-inputted values and stores them in variables, which are used to calculate the values for `total_cookies_needed`, `total_cookies`, `total_batches`, `trays_total`, `num_oven_runs_needed`, `mixed_dough`, and `trays_in_oven`. Note that the inputs `oven_temp` and `time_baking` are runtime measurements (sensor/timer readings) input by the user and implemented in Phase 6. The variables are:

- `ingredients_per_batch = [ingredient_0, …, ingredient_n]`: Stores the list of ingredient quantities for one batch
- `num_friends`: Stores the number of friends expected
- `cookies_per_friend`: Stores the number of cookies per friend
- `cookies_per_batch`: Stores the number of cookies produced from one batch

- `oven_slots`: Stores the maximum number of tray slots available in the oven
- `baking_time`: Stores the time needed to bake cookies
- `baking_temp`: Stores the temperature that the cookies need to cook

Runtime measurements (Phase 6.1 and 6.2.1)

- `time_baking`: Stores the timer reading for the set of trays baking in the oven
- `oven_temp`: Stores the temperature reading from the oven

3- The '`Initial Calculations`' phase contains several expressions used to calculate the different values needed to produce and bake the cookies. See the description of the previous set of Phases for variable descriptions. Note that the notation ⌈x⌉ refers to the ceiling function (round up). These calculations are:

- `total_cookies_needed = num_friends * cookies_per_friend`
- `total_batches = ⌈ total_cookies_needed / cookies_per_batch ⌉`
- `total_cookies = total_batches * cookies_per_batch`
- `trays_total = ⌈ total_cookies / cookies_per_tray ⌉`
- `num_oven_runs_needed = ⌈ trays_total / oven_slots ⌉`

4- The '`Mixed Dough`' phase implements the mixing loop that adds ingredient-by-ingredient from each of the dough batch ingredient lists to the `mixed_dough`.

- Increments the batch counter, it increments `i = i + 1` to keep track of the number of dough batches mixed, implements the '`One Dough Batch`', sets `n = 0`, and loops until `i = total_batches`
- 4.1- The '`One Dough Batch`' sub-phase loops through each ingredient of the dough batch `ingredients_per_batch` list and adds it to the `mixed_dough`. It adds each ingredient using an index counter `n` to keep track of each ingredient in the list, and after adding an ingredient, it increments `n = n + 1`. The Phase will loop until `n = ingredients_per_batch`

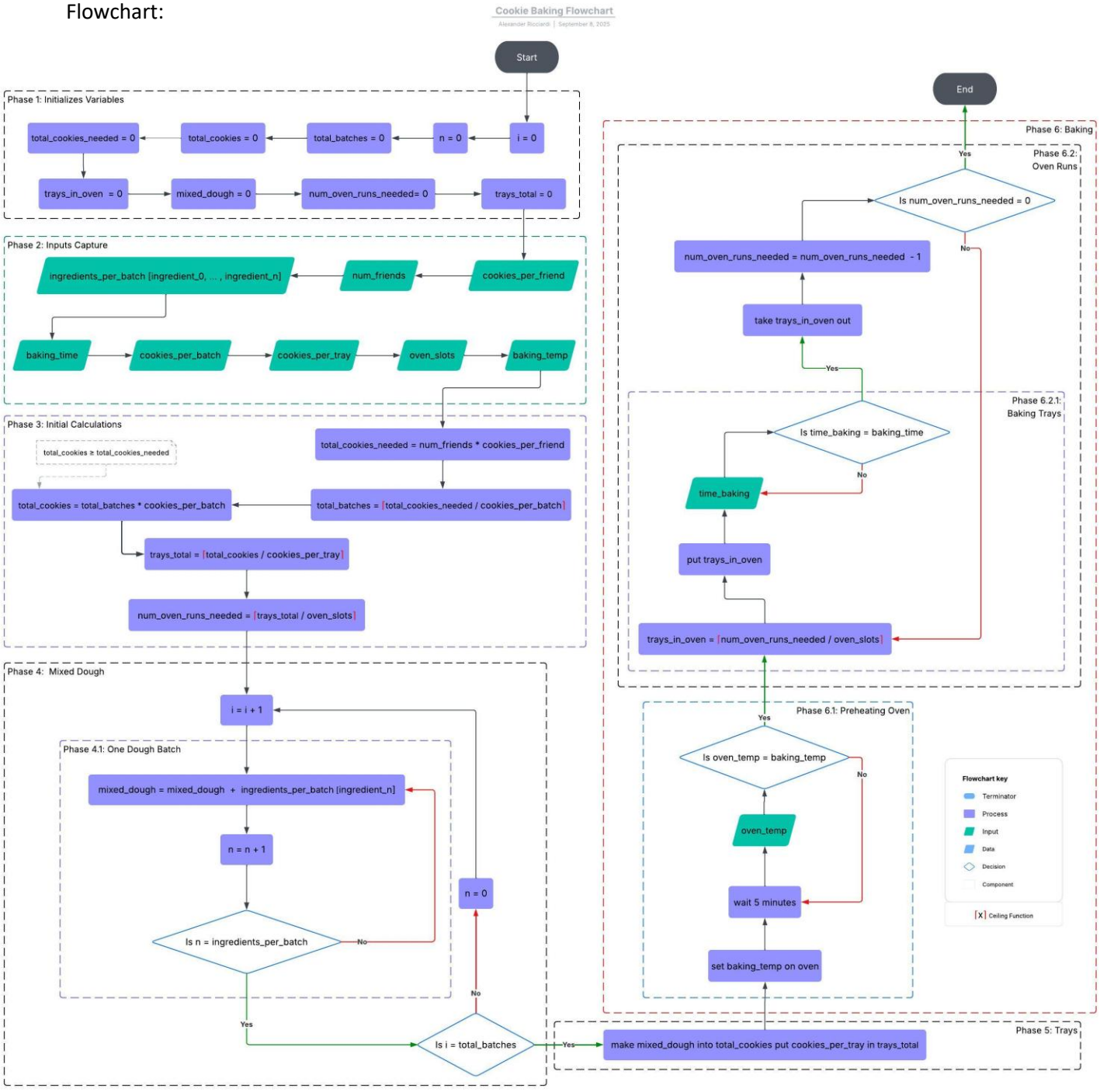
5- The '`Trays`' phase turns the mixed dough into individual cookies and places `cookies_per_tray` until the `trays_total` is reached. This Phase's logic description needs development

6- The '`Baking`' phase has nested sub-phases that implement preheating and repeated oven runs until all trays are baked

- 6.1- The '`Preheating Oven`' phase sets `baking_temp` on the oven, then loops, waits 5 minutes, and checks the `oven_temp` until `oven_temp = baking_temp`
- 6.2- The '`Oven Runs`' sub-phase loops oven cycles until no runs remain.
    - 6.2.1- It implements the '`Baking Trays`' sub-phase, which computes the number of trays that need to be put in the oven in this run based on the number of trays needing to be run and the number of oven slots: `trays_in_oven = ⌈num_oven_runs_needed / oven_slots⌉`, puts the trays in the oven, and bakes the trays until `time_baking = baking_time`
    - Then the '`Oven Runs`' phase takes the trays out of the oven, decrement `num_oven_runs_needed = num_oven_runs_needed − 1`
    - The phase loops until `num_oven_runs_needed = 0`

7- End -> cookies are read

Flowchart:



Food for thought:

Algorithms are sets of steps, tasks, tools, processes, or functionalities used to accomplish more complex tasks, implement complex functionalities, or solve complex problems. These sets can be referred to as modules or phases. In computer science, algorithms are a set of step-by-step processes used to perform tasks or solve problems; they are modularized mathematical processes (computation) expressed using code statements (expressions).

This task of baking cookies seems easy to us, but it is a very hard task for a Robot powered by AI to accomplish. We do not have AGI yet!
However, we need to remember that we have acquired the skills and the knowledge needed to bake cookies throughout our lifetime. In other words, these skills and knowledge are abstractions/modules (acquired through our lifetime), making it seem easy for us, but in reality, as shown by my flowchart, baking cookies is a complex process. Just give it time, and AI will eventually bake cookies faster and better than any human. All it needs is "Attention" with some tweaking to the transformer architecture.

-Alex