The “examples” listed were used in an attempt to comprehend the problem of calculating ratios for my factory in the context of gears.

Everything after the examples represents an outline of the calculator’s design.

Example Of Problem: Calculating Gear Ratios/Units Per Second Required To Satisfy Factory’s Current Needs

What do I want automated?

- Fast inserter per second
- Long inserter per second
- Normal inserter per second
- 2 x Assembler per base crafting time
- 2 x Electric mining drill per base crafting time
- 2 x ammo per base crafting time
- Hand supply of gears
    - Means producing slightly extra to fit inserter grab rate
- Hand supply of chips

Units being used:

- Gears: 1 gps per assembler
- Inserter can grab 0.85 gps
- gps : gears per second
    - == gpc/spc
- spc : seconds per craft
    - == gpc/gps
- gpc : gears per craft
    - == gps*spc

- uc : units crafted
    - == gr/gpc
- gr : gears required
    - == uc*gpc
- ups : units per second
- ar : assemblers required
    - == ups*spc

Gear ratios required for needed recipes:

```
/*
more of a placeholder to show that gpc is coming from
a gear requirement instead of a secondary craft
*/
```

(0) Gear @ 1 spc

(1) Inserter @ 1 spc

- ➢ **0** req @ 1 upc
- ■ 1 gpc
- ■ gps = 1 * ups

(2) Fast Inserter @ 1 spc

- ➢ **1** req @ 1 upc
- ■ 1 gpc
- ■ gps = 1 * ups

(3) Long Inserter @ 1 spc

- ➢ **0** req @ 1 upc
- ➢ **1** req @ 1 upc
- ■ 2 gpc
- ■ gps = 2 * ups

(4) Assembler @ 1 spc

- ➢ **0** req @ 5 upc
- ■ 5 gpc
- ■ gps = 5 * ups

(5) Electric Mining Drill @ 4 spc

- ➢ **0** req @ 5 upc
- ■ 5 gpc
- ■ gps = 1.25 * ups

Updated factory specs (output of calculator will look something like this):

1 @ 1 ups + (**2** req. 1 ups, **3** req. 1 ups)

- 3 ar
- 3 gps

2 @ 1 ups

- 1 ar
- ➢ **1** req @ 1 ups
    - 1 gps
- 0 gps (gps is already included in child recipe)

3 @ 1 ups

- 1 ar
- ➢ **1** req @ 1 ups
    - 1 gps
- 1 gps (half of gps is already included in child recipe)

4 @ 0.5 ups

- 0.5 ar
- 2.5 gps

5 @ 0.5 ups

- 2 ar
- 0.625 gps

Ratios Calculator Design Outline

Problem Statement

I want to know, given any combination of outputs (inputs in the context of the program, simply units per second of a series of items), what items I will need to produce in order to match these outputs as well as the # of these items per second, and the number of assemblers required to produce these items.

Inputs/Outputs Conceptualized

Input Info:

- Units per second of a given item I want to produce

Output Info:

- Units per second of ingredients required to make said item
- Crafter counts required to make said ingredients

All Starting Info Req. For Calculations

- Crafter speed multipliers (assembler, furnace, etc.)
    - a short decimal value
- Crafting time for each recipe
    - a short decimal value
- Ingredients for each recipe (link to a recipe)
    - an integer

Values Manipulated By The Program

These values are changed upon input, are used in calculations by the program, and will functionally serve as the program's outputs. Therefore, it'd be good to have some system that would prevent these values from becoming corrupted / make it easy to fix them if they do.

- Req. crafter counts for each item
    - a short decimal value
- Req. ups per item
    - a short decimal value

General Program Steps

1. accept an input value
2. store this input value
    a. I'm assuming this will just be done by default with the way sheets is setup

3. access the recipe the input value is referencing
    1. recipe will need to be looked up
    2. ingredients will need to be individually processed
4. use the recipe’s information to update/populate a visual output representing the factory’s required output to meet the input demand