

SEN-101 Algorithmic Thinking and Problem Solving
Assessment Activity SEN-101:00010
Turning Verbal Specifications into Algorithms
Last update: 14 August 2024

Assignment General Instructions

Below you will find the verbal specification for a problem. Read the specification carefully. Then create **both a set of flowcharts and pseudocode description** that correctly and completely express an algorithm for solving this problem.

Remember that algorithms must be written in terms of basic operations that a computer can perform. However, given the complexity of this problem, it is a good idea to use successive refinement to create multiple sub-flowcharts and pseudocode sub-process descriptions that describe the algorithm in natural pieces, plus a top-level algorithm that refers to these pieces. Show the top-level algorithm before the sub-processes.

Your flowcharts and the pseudocode should be **parallel in structure**. That is, when you break up the problem into sub-problems, both the flowcharts and the pseudocode should use the same sub-problem breakdown.

Your solution should consider all the issues we have discussed in the competency, including specifying and handling possible error conditions and boundary conditions. Failure to do defensive algorithm design will lower your score.

You should also follow the rules and guidelines for the two notations that we discussed in class. In particular, every flowchart sub-process should fit on one page with no off-page connectors, and flowcharts should never have lines that cross. Pseudocode should avoid using programming language terminology (such as “call function”, “declare variable”, etc.) except for logical keywords like “if”, “while”, etc.

Submit your work as a single PDF document that includes both your flowcharts and your pseudocode. Do not interleave the two notations; include your flowcharts first, then your pseudocode, or vice versa. Be sure that you include your full name and nickname at the top of the first page.

Embed your flowcharts as images. Be sure your diagrams are clear and readable. If the font is too small to read, this indicates that your flowchart is too complicated and should be decomposed into sub-flowcharts. Please do not submit hand-drawn flowcharts.

Each sub-problem (sub-flowchart and pseudocode section) must be on a separate page. Pages must have page numbers. Label each page with a title that explains the content of the page. If one flowchart or pseudocode section refers to another, use the page number to make the relationship clear.

If you do not follow instructions, you will get a lower score. If you submit only one notation (only flowcharts, or only pseudocode), I will not grade the assessment since my goal is to be sure that you are comfortable with both notations (and that you understand the relationship between them).

Problem Specification: Beverage Vending Machine

You are building a robotic beverage vending machine, rather like TaoBin.

Your machine will sell the following drinks:

- Espresso - 30 baht
- Americano - 30 baht
- Latte - 35 baht hot
- Cappucino - 35 baht hot
- Chocolate milk - 25 baht
- Cocoa - 40 baht
- Tea - 25 baht
- Cherry soda - 20 baht
- Peach soda - 20 baht

(We won't worry about hot versus cold drinks, or other variations.)

The primary function of the machine will be to repeatedly ask for the drink type, display the price, get the customer's payment, and deliver the drink, until the user enters a special code indicating the end of the day.

At the end of the day the machine will print the number of drinks of each type delivered and the total money received for all drinks.

If you've ever used a TaoBin machine, though, you know that sometimes it runs out of some ingredients. Then it becomes impossible to order some beverages. In our simple machine, we have limited amounts of two ingredients: coffee and milk. We will assume that the machine starts the day with 25 portions of each.

- The following drinks each use **one portion of coffee**: espresso, Americano, latte, cappucino
- The following drinks each use **one portion of milk**: latte, cappucino, cocoa
- Each order of chocolate milk uses **three portions** of milk.
- The other beverages do not use either ingredient.

The software that drives the machine must keep track of the remaining portions of each ingredient. If a user tries to order a drink but some ingredient is not available, the machine should reject that order.

Your algorithm must include the logic to check whether enough required ingredients are available and to reduce the amount available each time the machine uses some.

Suggestion: You may want to think about drinks falling into different categories: coffee drinks, milk drinks, and neither. Note that some drinks belong to both the coffee drink and milk drink categories. Consider that in the future, the owners of the machine may want to add new drinks, which might use one or both of the limited ingredients.