CS 310 | Fall 2020 | Unit 3 | Survey of Imperative Languages

Topics: *Imperative programming, language translation, subprograms*

Follow the instructions. Submit the indicated source files (📎) via the assignment in the eCampus lecture section by the due date on the course calendar. For each source file, include a comment with your full name and a brief statement acknowledging that your work complies with the academic integrity policy. The grading rubric is in this document (percentages are scaled as in the syllabus).

Background

The **factorial** of $n \geq 0$ is as follows: $n! = \begin{cases} 1 & \text{when } n \leq 1 \\ n(n-1)! & \text{otherwise} \end{cases}$

The number of **combinations** of r-many elements out of n-many elements without repetition is as follows: $\binom{n}{r} = \frac{n!}{r!(n-r)!}$

The **pancake problem** is as follows: *What is the maximum number of pieces which a pancake (a circle in 2 dimensions) can be cut into using n-many straight cuts?* The answer is $\binom{n}{0} + \binom{n}{1} + \binom{n}{2}$ pieces.

The **cake problem** is as follows: *What is the maximum number of pieces which a cake (a pancake extended to 3 dimensions) can be cut into using n-many straight cuts?* The answer is $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3}$ pieces.

The **hypercake problem** is as follows: *What is the maximum number of pieces which a hypercake (a cake extended to $k > 3$ dimensions) can be cut into using n-many straight cuts?* The answer is $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \cdots + \binom{n}{k}$ pieces.

Instructions

Choose **two** of the three language tasks below and follow the steps for your chosen language tasks (**50% each × 2**).

Scripting Language Task

A Choose **one** of these scripting languages: **JavaScript** or **Python**

B Code the program 📎 of the following steps in your chosen scripting language.

- 1 Implement the subprogram factorial(n) which returns $n!$ exactly as defined in the background.
- 2 Implement the subprogram combinations(n, r) which returns $\binom{n}{r}$ exactly as defined in the background.
- 3 Implement the subprogram hypercake(n, k) which returns the solution to the hypercake problem (number of pieces) for n-many cuts in k-many dimensions exactly as defined in the background.
- 4 Prompt the user for a number of cuts (n) and a number of dimensions (k) for the hypercake problem.
- 5 Display the solution to the hypercake problem for those parameters by calling hypercake accordingly.

C Nest the subprograms such that each is restricted to the scope of its caller and only one is in the global scope.

Compiled Language Task

D Choose **one** of these compiled languages: **Go** or **Rust**

E Code the program 📎 of step B in your chosen compiled language.

Domain-Specific Language Task

F Choose **one** of these domain-specific languages: **Pascal** or **Tcl**

G Code the program 📎 of step B in your chosen domain-specific language.

Bonus Opportunity

The following bonus opportunity is available.

H Complete **three** language tasks instead of just the required two.

I Clearly indicate which one of your language tasks you wish to be considered as bonus.

- 1 The indicated bonus language task is worth **25% bonus** added to your grade for the unit (maximum of 105%).
- 2 Any excess bonus above the maximum grade for this unit is redistributed to another eligible unit.