Student's name: _____

Homework 01: Computational Thinking with Functions

Introduction

The goal of this homework is to give you practice breaking down tasks into groups of computation (i.e., functions) with a non-python "language". You will use this language to produce patterns combining simple shapes.

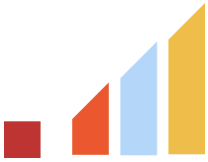
The Shape Language

Like Python, the shape language has basic types and operations.

Types

Our language is made of **four types** of shapes, that are

- 1 square and
- 3 trapezoids: short, medium, long.



Operations

Our language has one operation: which is to place a shape at some particular **location** in a 16x16 grid in some particular **orientation**. To perform this operation, you can call the following function:

```
putShape(shape, location, [orientation])
```

Where:

- `shape` is one of our four shapes, such as `square`, `medium trapezoid`, ...
- `location` is an actual `(x, y)` coordinate values and
- `orientation` is one of the label `NSEW+-` but is `[optional]`, given that square has no orientation.

Homework overview

This homework has three parts:

1. Practising using the language
2. Writing "code" to create a given pattern
3. Designing a new pattern, and writing code to create this new pattern

Submission instructions

Print this booklet, write the answers in the booklet and submit it to your instructor.

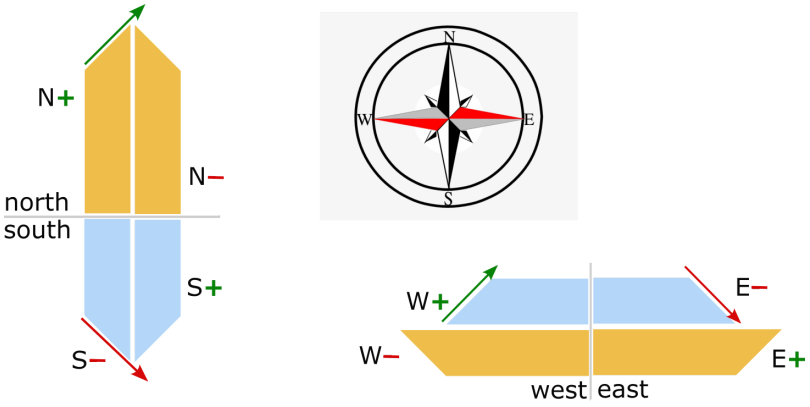
Part 1: Practicing using the language

1.1 - Specifying orientation

The trapezoids are directed in one of **eight orientations**. Their direction is either

- north or south (N/S), when placed vertically or
- west or east (W/E), when placed horizontally;

And for each of these compass directions, there are two versions depending of the slope of the trapezoid slanted line: + for a positive slope, − for a negative slope.



Warm-up: Within each of the 8 trapezoids below, write their orientation label, which are either N, S, W, or E each with a + or −.

1.2 - Specifying location

With these geometrical shapes, the computation is to place them in a pattern on the screen.

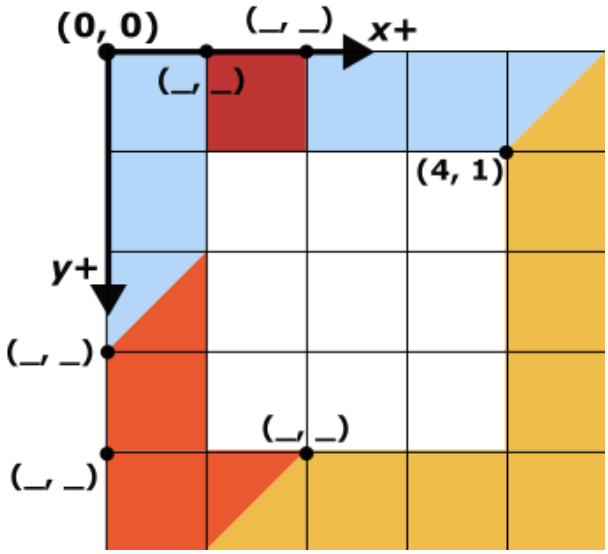
To do so, we define for each pattern configuration the **world coordinate system** with

- its origin at the top-left corner,
- the *x*+-axis going right, and
- the *y*+-axis going down.

Then each shape placement is specified with its top left corner coordinates within the world system.

As shown below for the seven shapes, two of those placements are identified within the world system. There are

1. the medium trapezoid (S+) that is placed at `(0, 0)`-coinciding with the system origin
2. the long trapezoid (N+) with its top left coordinates at `(4, 1)`.

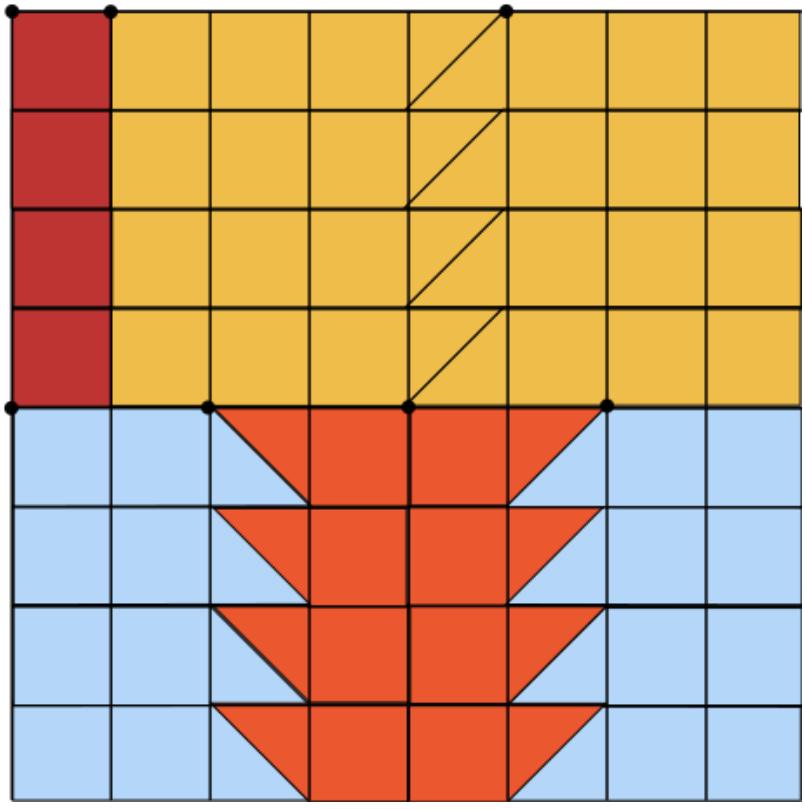


Warm-up 2: In the figure above, write

1. *the orientation label within each of the six trapezoids and*
2. *the coordinate values for the placement of each of the five shapes that haven't been provided.*

Part 2: Writing "code" for a provided pattern

Your task is to recreate the pattern below.



2.1 - Coding using only `putShape()`

Let's start by recreating the pattern using only the `putShape()` function, defined on page 1.

Planning

Task 1: Label the image above to understand the pattern, by writing

1. *the orientation label inside each shape and*
2. *the coordinates for their top left corner to determine their placement on the screen.*

As a guide, black points in the image indicate the top-left corner for one instance of each shape and orientation combination.

How many times will you need to call the `putShape()` function to re-create this pattern?

Coding

Task 2: Write below the exhaustive list of function calls that produces the pattern.
Also include comments in this code that explain what part of the pattern a single line or a group of lines are intended to produce

2.2 - Re-factoring with functions

The code you wrote above was very repetitive. In this part you will re-factor it. To do so, you first analyze groups found in the pattern, making important observations that serve to encode each. With this initial planning you then define **new** functions, that can be called to re-factor your solution.

Planning

Task 3.a: Identify some groups of shapes that are repeated (i.e., sub-patterns). You can draw them out, or describe them in English.

*Task 3.b: For each sub-pattern, what are the shape type(s) and orientation(s)? Give a **meaningful name** to represent each group that you will use throughout.*

Task 3.c: For each sub-pattern, where in the 16x16 grid do they need to be placed? I.e. provide the top-left corner `(x, y)` location for each sub-pattern.

To write code that will re-create each of these sub-patterns, you will define a function that will consume some parameters.

Task 3.d: For each function that draws a specific sub-pattern, specify the parameters that function will take.

Coding

You are now ready to define a few functions that will help you effectively code the overall pattern. Then you will rewrite your solution by writing a `main` that contains a series of calls to your new functions.

Task 4.a: For each of the sub-patterns you identified above, write code for a function that can generate it. You should define a few functions, i.e. for each its header and body. Reuse your meaningful names.

Task 4.b: For each of the sub-patterns, write calls in the `main` that will actually generate them. You should provided actual values for your new function parameters.

```
function main
    ...
```

Note: if you notice that the number of lines of code in *Task 4* is the same or more than in *Task 2*, something is wrong!

Part 3: Creating your own pattern

In this part you will create your own pattern and then write code to create the pattern.

3.1 - Understanding the materials provided

The last 2 pages (each double sided) provide *materials* for you to explore with and use to create your own pattern. In the first double-sided page, there are grids for you to place and paste cut shapes or draw the pattern you create for this final task. On the second double-sided page, on either side, there are

- a pattern made of four quadrants that are similar but not exactly the same, and
- the 28 shapes.

You should detach this second double-sided page and cut the shapes to make your collage on a grid.

3.2 - Create your pattern

Create your own pattern, ensuring that the pattern:

1. uses all the 28 shapes
2. uses at least half the orientations and
3. has at least 4 sub-patterns.

Task 5.a: *Include the pattern you created in this booklet, that you submit.*

Task 5.b: *Describe in a few words the sub-patterns that make your collage.*

3.3 - Code your pattern

Task 5.c: *Write code that will re-create the pattern that you designed.*