Contest 2.2: Beautiful Runes

Start date: 03 September 2017
Due: 11 September 2017, 23:59

Readings:

- Textbook Sections 1.1.1 to 1.1.4

Background

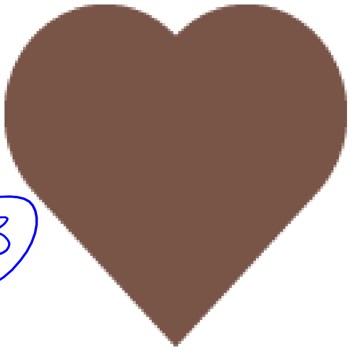
The Grandmaster has decided to have a new carpet installed in his office, but can't think of a suitable design. Immediately, you see your fellow Cadets springing to action, weaving intricate designs to impress the Grandmaster. The Grandmaster, understanding the importance of crowdsourcing, decides that the best design will be used for his carpet. You realise that you have nothing to lose by taking part in the impromptu competition. Who knows, if your design was chosen, it would be immortalised as a carpet in the Grandmaster's office.

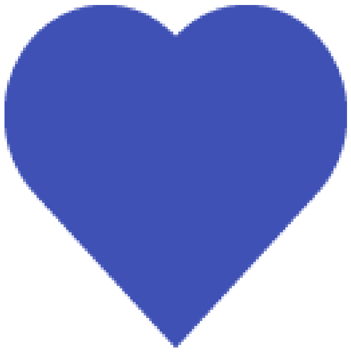
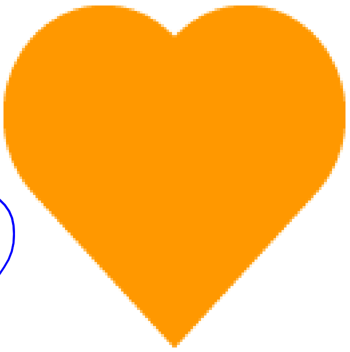
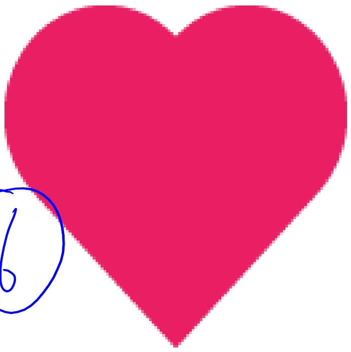
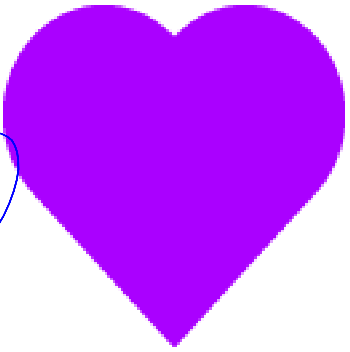
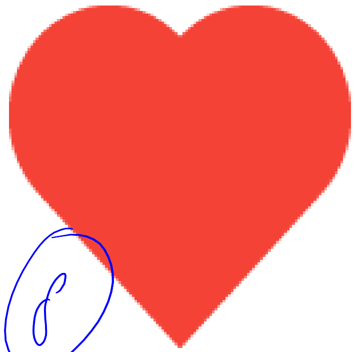
Task 1:

The Grandmaster strongly feels that carpet designs should not be monochrome. He brings out a colour palette, filled with a selection of colours. You seem to think that those are the Grandmaster's favourite colours.

You can now use the following function to create a new rune with the desired colour, beside `black(heart_bb)` and `white(heart_bb)`:

1 0

Colour	Reference Image	Colour	Reference Image
2 `blue`		3 `brown`	

Colour	Reference Image	Colour	Reference Image
4 `green`		`indigo`	
5 `orange`		6 `pink`	
7 `purple`		8 `red`	
9 `yellow`			

For those who like surprises, a special color function called `random_color` is also given to you. Nobody knows which color it will transform the rune into, but certainly not the boring black and white.

An example on how to use the new colour functions:

```
var pic = purple(heart_bb);
show(make_cross(pic));
```

would display a purple `heart_bb`.

You may submit up to 3 separate 2D runes. Each rune should be submitted as its own function (following the naming convention described below) such that running the function will return a rune that can then be displayed with `show`. Follow the given template:

```
// First entry
function yourname_2d_contest_0() {
}

// Second entry
function yourname_2d_contest_1() {
}

// Third entry
function yourname_2d_contest_2() {
}
```

More information on the contest, such as judging criteria, will be provided as the due date approaches. You are highly encouraged to only finalize your submission after those details have been published.

Rules for submission

- Be creative.
- No call to show, clear and any other attempt to interact with the viewport is allowed.
- Strictly follow the naming convention outlined in the contest PDF description.
- All functions should be self-contained.

Example of what is acceptable

```
function martin_henz_2d_contest_0() {
    function favorite(index) {
        return index === 0 ? heart_bb : black_bb;
    }
    return beside(favorite(0), favorite(1));
}
```

Example of what is not acceptable

```
function helper() {
    return star_bb;
}

// Not following naming convention
```

```
function my_2d_rune() {
    clear(); // Interacting with the output
    show(helper()); // Calling functions not defined within your entry
}
```

Evaluation of Submissions

- This contest (2.2) is for the 2D category of the rune contests.
- For each category, each student will be randomly assigned to 9 or 10 entries for voting. (This is to ensure that each submission receives an equal number of votes.)
- Each student can assign a score of 1 to 10 for each entry. The scores assigned to all 10 entries must sum up to 55. (The simplest way to achieve this is to rank the submissions from 1 to 10.) For students that are assigned 9 entries only, the total scores must sum up to 54.
- The final score for a submitted entry is `score(v,t) = v - 2^{t/50}`, where v is the normalized_voting_score (max 100), and t is the number of tokens in your program, including semicolons, operators, parentheses, but not including comments. The "acceptable" example above has 36 tokens.
- `normalized_voting_score = sum_of_scores / number_of_voters / 10 * 100`

Submission

To submit your work for this mission, copy the url on your browser and submit the url (there is a tab to submit a url; don't submit a file containing the url) to the IVLE Contest 2.2 submission folder. Strictly follow to the deadlines set at the start of this file.

IMPORTANT: Make sure that everything for your programs to work is on the left hand side and not in the interpreter! This is because only that program is preserved when opening the url you have given us.