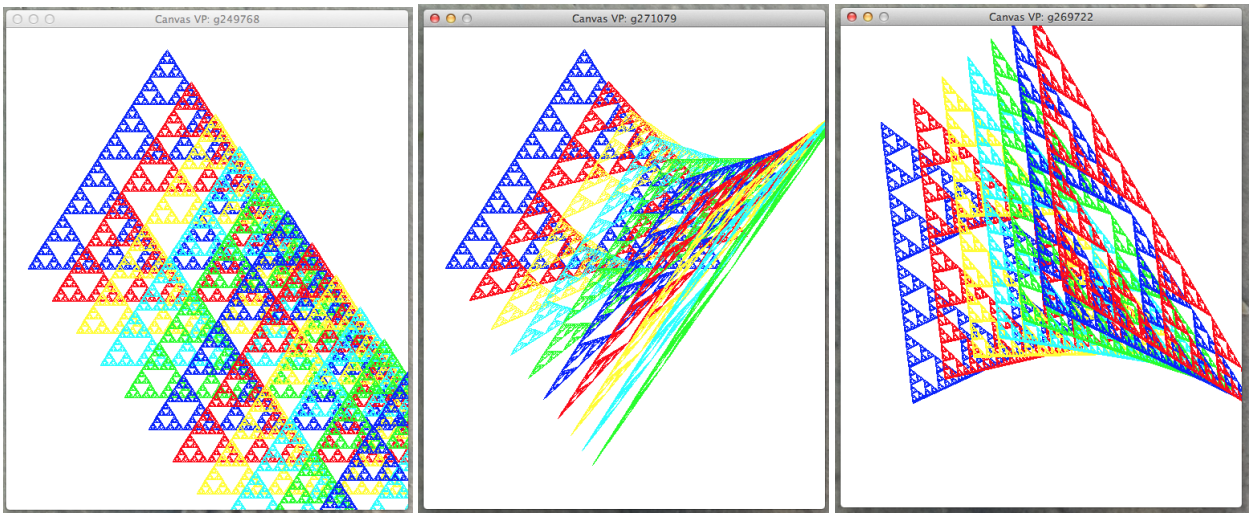
Exercícios Extra-Classe – Laboratório 5

IMPORTANTE: No arquivo com os exercícios abaixo, deixe no mínimo 3 testes não triviais para cada programa na janela de definições.

1. Considere as definições de dados e funções desenvolvidas no Laboratório 5. Construa o programa `triangulos1` que gera várias triângulos de Sierpinski , mudando a cor, conforme o exemplo na figura (a) abaixo. A entrada para o programa deve ser um triângulo (uma estrutura contendo os três vértices do triângulo), um número, representando a cor do primeiro triângulo a ser desenhado, e os deslocamentos nos eixos `x` e `y`. A chamada que gerou a figura (a) foi

   ```
   (triangulos1 (make-triangle A B C) 1 30 40)
   ```

   onde `A`, `B` e `C` são as constantes definidas no template do Laboratório 5 (antes de gerar a figura, deve-se abrir uma tela, poe exemplo, com `(start 500 600)`). A condição para término é que o primeiro vértice do triângulo tenha saído da tela.



(a) (b) (c)

2. Sobre o programa `triangulos1` responda as perguntas a) a e) do exercício 4 do Laboratório 5.

3. Construa agora o programa `triangulos2`, para gerar figuras como (b) e (c) acima. A cada nova iteração, o deslocamento diminui em 10% e o desclocamento da coordenada y do terceiro vértice do triângulo é negativo (ou seja, o deslocamento deve ser subtraído desta coordenada, ao invés de somado). A condição para término é que o valor absoluto do deslocamento no eixo y seja menor ou igual a 15. Para gerar as figuras (b) e (c), depois de gerar as telas, foram executados os seguintes comandos (novamente, considerando as constantes `A` a `F` do template):

   ```
   (triangulos2 (make-triangle A B C) 1 30 40)
   (triangulos2 (make-triangle F D E) 1 40 -30)
   ```

4. Desenvolva uma **função recursiva** `transforma` que, dada uma lista de símbolos contento nomes de seleções e a palavra 'Fim, gera uma lista de seleções, confome o exemplo abaixo:

```
 (define LISTA (list 'Brasil 'Argentina  'Chile 'Uruguai 'Fim
                     'Argentina 'Chile 'Brasil 'Fim
                     'Chile  'Paraguai 'Brasil 'Brasil 'Argentina 'Fim))
;;   (transforma LISTA) = (list
;;                     (make-seleção 'Brasil (list 'Argentina 'Chile 'Uruguai))
;;                     (make-seleção 'Argentina (list 'Chile 'Brasil))
;;                     (make-seleção 'Chile (list 'Paraguai 'Brasil 'Brasil 'Argentina)))
```

5. Argumente por que a função `transforma` sempre termina (lembre que, se forem usadas funções auxiliares, é necessário garantir que as funções auxiliares também terminam para assegurar que `transforma` termina).

6. Construa um programa que, dada uma lista de nomes de seleções, gera todos os confrontos entre elas. Cada confronto deve ser uma estrutura que contém os nomes das duas seleções. A ordem das seleções não é relevante. Por exemplo, se o programa receber as seleções Brasil, Argentina, Holanda e Uruguai devem ser gerados 6 confrontos: Brasil $\times$ Argentina (ou Argentina $\times$ Brasil), Brasil $\times$ Chile (ou Chile $\times$ Brasil), Brasil $\times$ Uruguai (ou Uruguai $\times$ Brasil), Chile $\times$ Argentina (ou Argentina $\times$ Chile), Uruguai $\times$ Argentina (ou Argentina $\times$ Uruguai), Chile $\times$ Uruguai (ou Uruguai $\times$ Chile).