Programming Languages
XX June 20XX

Rules

Time at disposal: 4 h.

Read carefully the following questions. For every question, provide an answer. Please remember to justify why the results hold by writing the logical flow that lead you to the answers.

Exercises

1. The following program is written using a pseudo-code. Assume that static scoping and call by reference are used. Expressions are evaluated from left to right. What does the following program print? Motivate the answer.

```
{
  int x = 0;

  int A(reference int y) {
         int x=2;
         y=y+1;
         return B(y)+x;
         }

  int B(reference int y){
        int C(reference int y){
              int x=3;
              return A(y)+x+y;
              }
        if (y==1) return C(x)+y;
        else return x+y;
        }
  write (A(x));
}
```

Assuming static scoping is implemented using a display. Show graphically the status of the display and of the activation record stack in the moment when the control enter for the second time into the function A. For every activation record, you can draw just the value of the field used to store the value of the previous display.

2. Describe in your own words what does call by name mean? What does it means that a variable is captured? How do we avoid capturing of variables?

   What does the following code print? Assume a language with static scoping and call by name (the `x++` returns the value of the variable `x` and then increments it). The specification of the language says that given an operator $\circ$, the valuation of the expressions $E_1 \circ E_2$ consist in: i) the valuation of E_1; ii) then the valuation of E_2; and the application of $\circ$ to the previous two values.

```
{
  int x=5;
  int P(name int m){
      int x=2;
      return m+x;
      }
  write(P(x++) + x);
}
```

   Motivate the answer.

3. Describe using your own words the *mark and sweep* technique: what this techniques is used for and what it does. In case the memory is an issue, what is the technique that can be used to perform the mark and sweep without wasting too much memory? What is the difference between mark and sweep compared to mark and compact? What are the pros and cons of these approaches?

4. Describe using your own words the notion of type equality and type compatibility. What does it mean structural type equivalence? What other alternatives are there for type equivalence?

   Describe a case of a programming language having two types T1 and T2 such that T1 is compatible with T2 but not equivalent.

5. Shortly describe how you decided to implement the data types used to encode the game configuration in your Onitama project and how you parse it from a string.

6. A lot of techniques for the compression of images are based on a tree data structure called "Quad Tree". Assume that the image is square and the size of the square is a power of 2. If the image is homogeneous (same color) it is encoded, regardless of its dimension, as a leaf containing its color (see Figure 1 for a graphical representation). If the image is not homogeneous, then it is encoded as a node whose child encode i) the upper left square, ii) the upper right square, iii) the bottom right square, and iv) the bottom left square using the data type

```
data QT a = C a | Q (QT a) (QT a) (QT a) (QT a)
```

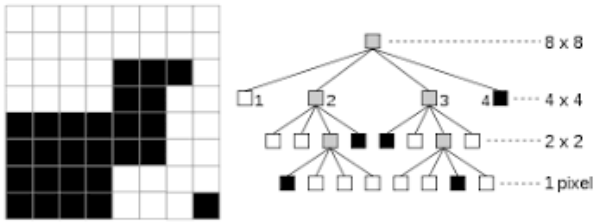

Figure 1: Graphical representation of a Quad Tree.

Write a function in Haskell named myFlip that given a list of Quad Trees returns the list of the flipped images respect to the horizontal axis. A flipped image or reversed image (the more formal term) is a static image that is generated by a mirror-reversal of an original across a horizontal axis (a flopped image is mirrored across the vertical axis).

Write the type signature of all the functions that you define.
Motivate your choices briefly describing your code.

7. Write a function zipWith for Quad Trees that, analogously to what happens for the function zipWith for list, given

   - a binary operation f on the elements of the Quad Trees
   - a Quad Tree q1
   - a Quad Tree q2

   builds the Quad Tree that encodes the image that is obtained by applying the function f to all the pixels of the image in the same position in the images encoded by q1 and q2.

   As an example

   ```
   let z = C 0; u = C 1; q = Q z u u u
   in zipWith (+) q (C 2)
   ```

   will produce

   ```
   Q (C 2) (C 3) (C 3) (C 3)
   ```

   Write the type signature of all the functions that you define.
   Motivate your choices briefly describing your code.

8. What does the identifier `Monad` denote in Haskell?

   The MonadPlus type class is defined as follows.

   ```
   class Monad m => MonadPlus m where
       mzero :: m a
       mplus :: m a -> m a -> m a
   ```

Lists are an instance of MonadPlus defined as follows.

```
instance MonadPlus [] where
    mzero = []
    mplus = (++)
```

Consider the following definition introducing the `guard` function.

```
guard :: (MonadPlus m) => Bool -> m ()
guard True = return ()
guard False = mzero
```

What does the following expression return? Motivate your answers showing the steps that Haskell does to produce the result.

```
guard (5 > 2) >> return "cool" :: [String]
```

And the following expression? Motivate your answers showing the steps that Haskell does to produce the result.

```
guard (1 > 2) >> return "cool" :: [String]
```

What does the following code return? Motivate your answer.

```
do
  x <- [1,2]
  y <- [1,2]
  guard $ x <= y
  return (x + y)
```