CIS 624 Structure of Programming Languages, Assignment 3
Due: Thursday, Nov 2, Bitbucket or Deschutes drop box

Make sure that you

- Understand the representations of numbers, booleans, pairs , etc. in the lambda-calculus
- Understand how recursion is represented using fix defined as $\lambda f.\ (\lambda x.\ f\ (\lambda y.\ x\ x\ y))(\lambda x.\ f\ (\lambda y.\ x\ x\ y))$

Problems

1. Write the lambda function for `"isEqual"` that tests two (Church) numbers for equality and return a Church boolean. For example, `"isEqual" "3" "3"` should return $(\lambda t.\ \lambda f.\ t)$ and `"isEqual" "3" "4"` should return $(\lambda t.\ \lambda f.\ f)$ (hint: you can use any of the other operations on Church numerals defined in lecture).

2. Recall $(\lambda x.\ e)\ e' \rightarrow e[e'/x]$ refers to the lambda expression obtained by replacing each *free* occurrence of the variable x in e by the lambda expression e'. Such a substitution is called **valid** or **safe** if no free variable in e' becomes bound as a result of the substitution $e[e'/x]$. An invalid substitution involves a **variable capture** or **name clash**.

   Relate through an example the "variable-capture" problem, i.e., show a case where variable capture occurs.

3. Create a combinator function and use it to compute the Ackermann function recursively. The Ackermann function is defined as follows.

$$\begin{aligned} A(0,n) &= n+1 \\ A(m+1,0) &= A(m,1) \\ A(m+1,n+1) &= A(m,A(m+1,n)) \end{aligned}$$

   DO NOT use the language's native support for recursion, i.e., if you use OCaml, you may not use `rec` in your function definition. **DO NOT** use a simple iteration (loop) instead of the combinator.

   You can use a language of your choice, as long as it's possible to test on ix. For examples of a Y-combinator in a variety of languages, see `http://rosettacode.org/wiki/Y_combinator`.

   Name your file `problem3.[suffix]` where `[suffix]` is the suffix corresponding to your implementation language, e.g., `.ml` for OCaml, `.py` for Python, etc.

What to turn in:

- Hard-copy (written or typed) answers to problems: Deschutes dropbox.
- For electronic submissions, please use your Bitbucket cis-624 git repository and place solution files in **hw3** subdirectory. No paper copy needed if you are submitting everything through Bitbucket.