

CS314 Principles of Programming Languages

Written Assignment 2

Name: _____

NetID: _____

Instructions:

- The written assignment can be completed in 1 hour.
- The quiz has a total of 8 points.

1. True or False

- (a) ($\frac{1}{2}$ point) In Prolog, $A+b$ unifies with $b+A$. T / F.
- (b) ($\frac{1}{2}$ point) Reordering the terms in the body of a Prolog rule may change the result. T / F
- (c) ($\frac{1}{2}$ point) The result of the query `?- 3 is A + 1 is A = 2.` T / F
- (d) ($\frac{1}{2}$ point) With `occurs_check` enabled, a Prolog query can avoid infinite search. T / F

2. What is a unifier of each of the following terms. Assume that `occurs-check` is true.

- (a) ($\frac{1}{2}$ point) $f(X,Y,Z) = f(Y,Z,X)$
- $\{X/Y, Y/Z\}$
 - $\{X/Y, Z/y\}$
 - $\{X/A, Y/A, Z/A\}$
 - None of the above.
- (b) ($\frac{1}{2}$ point) $tree(X, tree(X,a)) = tree(Y,Z)$
- Does not unify.
 - $\{X/Y, Z/tree(X, a)\}$
 - $\{X/Y, Z/tree(Y, a)\}$
 - $\{Y/X, Z/tree(Y, a)\}$
- (c) ($\frac{1}{2}$ point) $[A,B,C] = [(B,C),b,a(A)]$
- Does not unify.
 - $\{A/(b, a(A)), B/b, C/a(A)\}$
 - $\{A/(b, a(C)), B/b, C/a(A)\}$
 - None of the above.

3. (1 point) Fill in the implementation of the `segment(A,B)` predicate below, which holds when A is a contiguous segment contained anywhere within list B. For example:

```
?- segment([3,5], [1,2,3,4,5]).
false.
```

```
?- segment([X,Y], [1,2,3,4]).
X = 1, Y=2;
X = 2, Y=3;
X = 3, Y=4;
false.
```

```
?- segment([3,4,X], [1,2,3,4,5]).
X=5;
false.
```

Fill in implementation below (hint: use the prefix, suffix, and append functions you have learned and you do not need to provide the code of these functions).

4. We are going to encode a graph over cities in Prolog. In particular, `link(a,b)` represents the fact that there is a path from city `a` to city `b`. For example:

```
link(san_diego , seattle) .  
link(seattle , dallas) .  
link(dallas , new_york) .  
link(new_york , chicago) .  
link(new_york , seattle) .  
link(chicago , boston) .  
link(boston , san_diego) .
```

- (a) ($\frac{1}{2}$ point) First, write a predicate `path_2(A,B)` which holds if there is path of length two from `A` to `B`. The path is allowed to have duplicate cities. For example:

```
?- path_2(new_york , B) .  
B = boston ;  
B = dallas .
```

```
?- path_2(A , dallas) .  
A = san_diego ;  
A = new_york ;  
false .
```

Fill in your implementation of `path_2` below:

- (b) (1 point) Write a predicate `path_N(A, B, N)` which holds if there is a path of length `N` between `A` and `B`. The path is allowed to have duplicate cities, and you can assume that `N` is greater or equal to 1. For example:

```
?- path_N(new_york , B , 2) .  
B = boston ;  
B = dallas ;  
false .
```

```
?- path_N(new_york, B, 3).
B = san_diego ;
B = new_york ;
false .
```

```
?- path_N(A, san_diego, 5).
A = seattle ;
false .
```

Hint: you may want to consider two cases in your code: $N = 1$ and $N > 1$.

Fill in the implementation of path_N below:

5. (1 point) In this problem we will write a matrix transpose function in Python functions. A matrix is a two dimensional array, which we will represent as a list of lists of integers. For example, the following is a 2×3 matrix (meaning the height of the matrix is 2 and the width is 3):

```
A=[[ 1,  2,  3],
   [ 4,  5,  6]]
```

The transpose of a matrix A of dimensions $n \times m$ is a matrix B of dimensions $m \times n$ such that $A[i][j]$ is equal to $B[j][i]$ (for all valid indices i and j into matrix A). For example:

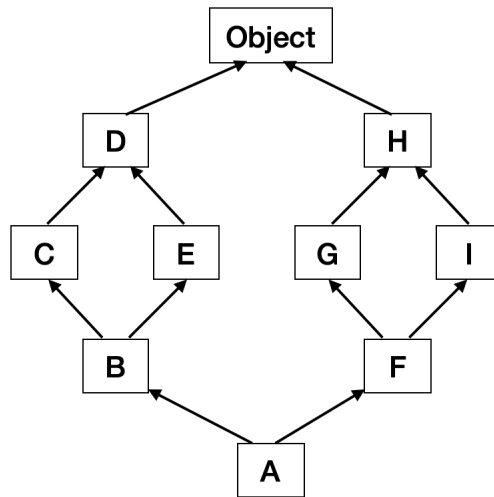
```
>>> transpose ([[ 1,  2,  3],
                [ 4,  5,  6]])
[[1, 4],
 [2, 5],
 [3, 6]]
```

Your code must be in this form:

```
def transpose(m):
    height = len(m)
    width = len(m[0])
    return [ [ _____ for ___ in _____ ] for ___ in _____ ]
```

Fill in the return statement of transpose below:

6. (1 point) Consider the following Python inheritance graph for classes *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H*, *I* and *Object*:



Fill in the method resolution order of class *A*.