BBM 301 - Programming Languages - Fall 2021 2nd Midterm
December 10, 2021

Name: _____

Student ID number: _____

Please write your name, ID and following honor pledge:
"**On my honor, I pledge that I have neither given nor received any unauthorized assistance on this exam.** "

"I give permission that my camera recordings will be taken for identification purposes for BBM301 exam and I understand that these recordings will not be used for any other purposes".

and **sign** your answer sheet.

1- Precedence in BNF [20 pts]

Consider a simple grammar for describing first order predicate logic with *connectives* ↔, →, ∧, ∨, and ¬; quantifiers ∀, ∃ and parentheses. The following rules are available in this BNF:

- An atomic expression is in the form of `predicate(variable_list)`, such as **`P(x)`**, or **`Q(x,y)`**. Variables in this list are limited to x and y.
- The precedence of the connectives from highest to lowest are ¬ , ∧, ∨, →, ↔.
- Quantifiers ∀, ∃ have precedence just below ¬, and higher than all the other four connectives.
- As always, you can use parentheses to change the priority.

For example,

`¬ ∃x ( P(x) → ∀y ¬ ( ∃z R(y , z) ∨ S(y) ∧ Q(x) ) )` is parsed as

`[¬ ∃x] [P(x) → [∀y [ ¬ [ [∃z R(y , z)] ∨ [S(y) ∧ Q(x)] ] ] ] ]`

a) In the following lines, the grammar is given in BNF. **Complete the grammar by filling in the blanks with the correct expression ID (exp1, exp2, exp3, exp4, or exp5)** for the grammar to satisfy the precedence order of the rules described above. Note that, all the operators are left associative. **Be careful, as the connectives and quantifiers are placed in random order.**

```
<atomic_exp> ::= <predicate> ( <var_list> )
<var_list> ::= <var> | <var_list> , <var>
<var> ::= x | y
<predicate> ::= P | Q | R | S

<exp> ::= ( <_____> ) | ¬ <_____> | <atomic_exp>

<exp1>      ::= ∀ <var> <_____>   | ∃ <var> <_____>

<exp2>      ::= <exp2>  ∧  <_____> | <_____>

<exp3>      ::= <exp3>  ∨  <_____> | <_____>

<exp4>      ::= <exp4>  →  <_____> | <_____>

<exp5>      ::= <exp5>  ↔  <_____> | <_____>
```

b) Construct the parse tree for the following sentence

`¬ ∃x ( P(x) → ∀y ¬ ( ∃z R(y , z) ∨ S(y) ∧ Q(x) ) )`

2- BNF for simple Prolog [20 pts]

A list in Prolog is either an empty list or a list of elements which can be atoms, atomic propositions, or other terms including lists. A list is enclosed in square brackets and elements of the list are separated by commas. Lists can also be represented using the Head and the Tail of the list as `[ Head | Tail ]`.

In this question, for simplicity, assume that elements are restricted to digits or other lists. Some possible lists are:

```
[ ]
[ 1 , 2 ]
[ 1 , [ ] , [ 2 , [ 3 , 4 ] ] ]
[ 1 | [ 2 | [ 3 | [ 4 , 1 ] ] ] ]
[ 1 , 2 , [ 3 | 4 ] | [ ] ]
```

Write a grammar using the BNF to describe lists in Prolog. Do not use EBNF notation.

3- Prolog [15 pts]

Consider the following database in Prolog.

```
male(tom).
male(jack).
male(bob).
male(mike).
male(david).
male(fred).

female(jane).
female(jill).
female(mary).
female(alice).
```

```
parent(jane, jill).
parent(jane, mary).
parent(tom, mary).
parent(tom, jill).
parent(jill, fred).
parent(jill, david).
parent(jack, fred).
parent(jack, david).
parent(mary, alice).
parent(mary, mike).
parent(bob, mike).
parent(bob, alice).
```

a) Write a rule **`same_gender(X,Y)`** for finding the people whose gender is the same.

b) Now, consider the following query which tries to find the people whose sibling has the same gender as himself/ herself.

```
parent(X,Y), parent(X,Z), same_gender(Y,Z), Y\==Z.
```

Write the first 6 answers to this query by instantiating the variables X, Y and Z with the correct constants in correct order.

	X=	Y =	Z=
1st			
2nd			
3rd			
4th			
5th			
6th			

c) Write a query that corresponds to the following question.

Whose mother is Alice's aunt?