

## CS 4303 Programming Language Concepts

### Project 2 – Logic Programming in Prolog

Write a Prolog program to evaluate prefix arithmetic expressions represented as lists. Assume four operators may be used in expressions and they are +, -, \*, and /. All these operators are binary, i.e., they are applied in an expression in this format: [operator, operand1, operand2], where “operator” can be +, -, \*, or /, and each operand can be either a single number, or another expression. For example, the following are all legal input:

```
[+, 1, 2]
```

```
[-, 6, [/ , 7, 0.5]]
```

```
[*, [/ , 78, [-, 67, 3.5]], [+ , 4, 9.0]]
```

But the following are not:

```
[+, 1] ; missing operand
```

```
[6, -, 7] ; every expression must start with an operator
```

```
[/, +, 4, 5] ; every expression can only have one operator
```

```
[*, 6, 7, 8] ; every expression must have exactly two operands
```

```
[% , 6, 7] ; unrecognizable operator
```

```
[-, 6, A] ; illegal operator
```

Write a predicate named “main” to check the validity of the expression and evaluate the expression. The following should be the way to run your program:

```
➤ main([+, 1, 2], X)  
X = 3
```

```
➤ main([- , 6, A], X)  
error
```

where [+ , 1, 2] and [- , 6, A] are input. If the input is legal, your program should output the final result of the expression evaluation, as shown in the above example. If the input is illegal, the result is undefined, i.e., your program assumes the input is legal.

Submit your program file at Blackboard Vista by the due date (Check the course schedule).