

1) For each of the following sets of strings, give a BNF grammar, an automaton diagram, and a regular expression which represents it

- a) all strings of 0s and 1s (binary strings) which end in 01
- b) all integers (in the usual decimal/base-10 notation) that are evenly divisible by 5

2)

Consider the following BNF-type grammar. (Note: the _'s indicate a blank space.)

{Note the bolded parentheses on the next line. Those are the CHARACTERS '(' and ')'}

```
<Expression> -> <Number> | ( <List> ) |
<Symbol> | <Operator> | <QuotedItem>
```

```
<Operator> -> + | - | * | /
```

```
<Symbol> -> <Letter> { <Letter> | <Digit> }*
```

```
<Number> -> <Digit> { <Digit>}*
```

```
<Digit> -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

```
<Letter> -> A | B | ... | Y | Z | a | b | c | ... | y | z
{I think you can get the idea on this one, even with
the ...'s .}
```

```
<List> -> [ <Expression> {<Spaces> <Expression>}*
]
```

```
<Spaces> -> _ { _ }*
```

```
<QuotedItem> -> ' <Expression>
```

For each of the following, determine if it is an “Expression”. Turn in a sheet which has the answer for each listed. Choose 3 of those that are valid Expressions and draw the parse tree for them.

8	891	-891	(8_str)
()	'(8_str)	hello4	4hello
(8+_5)	(6_-_var)	var-6	

