

## Grammar #1:

$$\langle \text{sign} \rangle = '+' \mid '-'$$
$$\langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
$$\langle \text{letter} \rangle ::= A \mid B \mid C \mid D \mid E \mid F$$
$$\langle \text{hex} \rangle ::= \langle \text{sign} \rangle \text{'0x'?} \langle \text{digit} \rangle \mid \langle \text{letter} \rangle \mid \langle \text{hex} \rangle \langle \text{digit} \rangle \mid \langle \text{hex} \rangle \langle \text{letter} \rangle$$
$$\langle \text{Addition} \rangle ::= \langle \text{hex} \rangle + \langle \text{hex} \rangle$$
$$\langle \text{Subtraction} \rangle ::= \langle \text{hex} \rangle - \langle \text{hex} \rangle$$

### Grammar #2:

$$\langle \text{program} \rangle ::= \langle \text{stmt-list} \rangle$$
$$\langle \text{stmt-list} \rangle ::= \langle \text{stmt} \rangle \mid \langle \text{stmt-list} \rangle$$
$$\langle \text{stmt} \rangle ::= \langle \text{identifier} \rangle = \langle \text{expr} \rangle$$
$$\langle \text{variable} \rangle ::= \text{"VAR"} \langle \text{identifier} \rangle$$
$$\langle \text{identifier} \rangle ::= \langle \text{id character} \rangle \mid \langle \text{identifier} \rangle \langle \text{id character} \rangle \mid \langle \text{identifier} \rangle \langle \text{decimal digit} \rangle$$

**<id character> ::= <letter>**

$$\langle \text{letter} \rangle ::= A|B|C \dots |Z|a|b|c \dots |z$$
$$\langle \text{decimal digit} \rangle ::= 0|1|2|3|4|5|6|7|8|9$$
$$\langle \text{integer} \rangle ::= [-] \langle \text{decimal digit} \rangle \mid \langle \text{integer} \rangle \langle \text{decimal digit} \rangle$$

```
<string> ::= ' ' <characters> ' ' | "
```

$$\langle \text{characters} \rangle ::= \langle \text{character} \rangle \mid \langle \text{characters} \rangle \langle \text{character} \rangle$$

```

<character> ::= <character> | <character>
<character> ::= ' | {any EBCDIC character}

```

```

<character> ::= | {any EBCDIC character}
<expr> ::= <term> | <term><operator><term> | <string>

```

`<term>` ::= `<integer>` | `<decimal digit>` | `<identifier>` | `<letter>`

<b>&lt;term&gt;</b>	<b>::=</b> <b>&lt;integer&gt;</b>
<b>&lt;operator&gt;</b>	<b>::=</b> <b>+</b> <b> </b> <b>-</b> <b> </b> <b>*</b> <b> </b> <b>/</b> <b>**</b>

```

<addition> ::= <term>+<term>

```

$$\langle \text{subtraction} \rangle ::= \langle \text{term} \rangle - \langle \text{term} \rangle$$
$$\langle \text{multiplication} \rangle ::= \langle \text{term} \rangle^* \langle \text{term} \rangle$$

<division> ::= <term>/<term>

$$\langle \text{exponentiation} \rangle ::= \langle \text{term} \rangle^{**} \langle \text{term} \rangle$$

```
<print> ::= PRINT <OBJ>
```

```
<OBJ> ::= <variable>|<string>|<integer>
```

RegEx:

1.

$$\text{VAR}\backslash s[a-zA-Z]+[a-zA-Z0-9]^*$$

2.

-?[0-9]+

3.

$$^{\wedge} "[a-zA-Z\s.]*" \$$$

4.

$$[a-zA-Z]+[a-zA-Z0-9]^*=[a-zA-Z][a-zA-Z0-9]^*[a-zA-Z\backslash s.]^*"-?[0-9]+$$

5.

To match the five operators

$$[\neg + \neg - \vee] [\neg^*] \{1, 2\}$$

To match the operands(variables/numeric values) with the five operators

$$/([a-zA-Z]+[a-zA-Z0-9]*|[0-9])([+|-|\w|[*]{1,2})([a-zA-Z]+[a-zA-Z0-9]*|[0-9])/s$$

6.

To match the PRINT keyword

^PRINT

To match the object(variable, numeric value, string) after PRINT keyword

```
^PRINT\s("[a-zA-Z\s.]"*|VAR\s[a-zA-Z]+[a-zA-Z0-9]*|[0-9]+)
```