

Grammar

$P \rightarrow \{ \{S\} \}$	$\{ \{ \{ \} \}$
$S \rightarrow A \mid G \mid O \mid C \mid W$	$\{\text{let}, \text{read}, \text{print}, \text{if}, \text{while}\}$
$A \rightarrow \text{let ID} := E ;$	$\{\text{let}\}$
$E \rightarrow B \{ (\text{and} \mid \text{or}) B \}$	$\{\text{not}, -, (, \text{ID}, \text{FLOATLIT}\}$
$B \rightarrow R [(< \mid > \mid ==) R]$	$\{\text{not}, -, (, \text{ID}, \text{FLOATLIT}\}$
$R \rightarrow T \{ (+ \mid -) T \}$	$\{\text{not}, -, (, \text{ID}, \text{FLOATLIT}\}$
$T \rightarrow F \{ (* \mid /) F \}$	$\{\text{not}, -, (, \text{ID}, \text{FLOATLIT}\}$
$F \rightarrow [\text{not} \mid -] U$	$\{\text{not}, -, (, \text{ID}, \text{FLOATLIT}\}$
$U \rightarrow (E) \mid \text{ID} \mid \text{FLOATLIT}$	$\{(, \text{ID}, \text{FLOATLIT}\}$
$G \rightarrow \text{read} [\text{STRINGLIT}] \text{ID};$	$\{\text{read}\}$
$O \rightarrow \text{print} [\text{STRINGLIT}] [\text{ID}];$	$\{\text{print}\}$
$C \rightarrow \text{if} (E) P [\text{else } P]$	$\{\text{if}\}$
$W \rightarrow \text{while} (E) P$	$\{\text{while}\}$

Error conditions (display an appropriate message and quit execution):

1. Syntax error
2. Use of an uninitialized variable

Simple Example 1:

```
{
  let i := 1;
  let j := 5;
  read "input a number: " k;
  let t := -k;
  print "negative of k is: " t;
  let t := 1+t*-j+k;
  print "output is: " t;
}
```

Simple Example 2:

```
{
  let i := 2.5;
  let j := 5.0;
  if (i > j)
  {
    let k := 2 * i + j;
    print "k: " k;
  }
  else
  {
    let k := i * j + 2;
    print "k: " k;
  }
}
```

Nested If Example:

```
{
  let i := 2.5;
  let j := 5.0;
  if (i > j)
  {
    let k := 2 * i + j;
    print "k: " k;
    if ((k < 200) and (k > 0))
    {
      print "k is between 200 and 0.";
    }
    else
    {
      print "k is not between 200 and 0.";
    }
  }
  else
  {
    let k := i * j + 2;
    print "k: " k;
    if ((-k > -200) and (-k < 0))
    {
      print "-k is between -200 and 0.";
    }
    else
    {
      print "-k is not between -200 and 0.";
    }
  }
}
```

Simple While Example

```
{
  let i := 0.0;
  while (i < 5)
  {
    print "i: " i;
    let i := i + 1;
  }
}
```

Nested While Example

```
{
  let i := 0.0;
  while (i < 5)
  {
    print "i: " i;

    let j := 0;
    while (j < i)
    {
      print "j: " j;
      let j := j + 1;
    }
    print;
    let i := i + 1;
  }
}
```

Factorial Calculator:

```
{
  let t := 1;
  print "k! calculator.";
  read "input the value of k: " k;
  if (k < 0)
  {
    print "The value of k cannot be negative."
    print "You entered: " k
  }
  else
  {
    while (k > 1)
    {
      let t := t * k;
      let k := k - 1;
    }
    print "k! is: " t;
  }
}
```