**Output (starting with <pgm> rule). This is the required version. You can optionally choose to create a start rule and have it call <pgm>.**

Welcome to DrRacket, version 6.8 [3m].

Language: plai, with debugging; memory limit: 128 MB.

> (pgm)

Entering <stmt>

Entering <assign>

Entering <id>

Found x

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Entering <int>

Found 3

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Entering <stmt>

Entering <assign>

Entering <id>

Found y

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Entering <int>

Found 5

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Entering <stmt>

Entering <assign>

Entering <id>

Found z

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Found (

Entering <exp>

Entering <term>

**Interactions disabled 🡨 *this will randomly appear somewhere due to when program exits vs what has already been printed***

Entering <factor>

Entering <id>

Found x

Leaving <id>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Found +

Entering <term>

Entering <factor>

Entering <id>

Found y

Leaving <id>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <etail>

Leaving <exp>

Found )<Leaving factor>

Entering <ttail>

Found /

Entering <factor>

Entering <int>

Found 3

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Entering <stmt>

Found print

Entering <exp>

Entering <term>

Entering <factor>

Entering <id>

Found z

Leaving <id>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <stmt>

Entering <stmt>

Entering <assign>

Entering <id>

Found x

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Entering <id>

Found abc

Leaving <id>

Leaving <factor>

Entering <ttail>

Found \*

Entering <factor>

Entering <int>

Found 3

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Reached end of file. Parse complete...

Parse completed with 0 errors.

Now exiting program

**Output (using <start> rule to call <pgm>): Not required. I will include for the sake of being thorough and correct, but since I didn’t specify, I won’t require anyone to add this.**

Welcome to DrRacket, version 6.8 [3m].

Language: plai, with debugging; memory limit: 128 MB.

> (start)

Entering <pgm> rule

Entering <stmt>

Entering <assign>

Entering <id>

Found x

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Entering <int>

Found 3

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Entering <stmt>

Entering <assign>

Entering <id>

Found y

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Entering <int>

Found 5

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Entering <stmt>

Entering <assign>

Entering <id>

Found z

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Found (

Entering <exp>

Entering <term>

Entering <factor>

Entering <id>

Found x

Leaving <id>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Found +

Entering <term>

Entering <factor>

Entering <id>

Found y

Leaving <id>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <etail>

Leaving <exp>

Found )<Leaving factor>

Entering <ttail>

Found /

Entering <factor>

Entering <int>

Found 3

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Entering <stmt>

Found print

Entering <exp>

Entering <term>

Entering <factor>

Entering <id>

Found z

Leaving <id>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <stmt>

Entering <stmt>

Entering <assign>

Entering <id>

Found x

Leaving <id>

Found =

Entering <exp>

Entering <term>

Entering <factor>

Entering <id>

Found abc

Leaving <id>

Leaving <factor>

Entering <ttail>

Found \*

Entering <factor>

Entering <int>

Found 3

Leaving <int>

Leaving <factor>

Entering <ttail>

Next token is not \* or /, choosing EPSILON production

Leaving <ttail>

Leaving <ttail>

Leaving <term>

Entering <etail>

Next token is not + or -, choosing EPSILON production

Leaving <etail>

Leaving <exp>

Leaving <assign>

Leaving <stmt>

Reached end of file. Parse complete...

Parse completed with 0 errors.

Exiting <pgm> rule

Now exiting program

**Interactions disabled 🡨 *this will randomly appear somewhere due to when program exits vs what has already been printed***