

Enter one or more lines. End with a single line with \$  
If you want to stop testing, enter EXIT

ROTATE 3

\$

The input was accepted by the lexer.

Token Stream: R#?

Strings: [ROTATE, 3,\n]

Parser result: accepted

-----

---

Enter one or more lines. End with a single line with \$

If you want to stop testing, enter EXIT

```
ROTATE 3
```

```
IF BLOCKEDP [MOVE 1
```

```
NOP]
```

```
$
```

The input was accepted by the lexer.

Token Stream: R#???w[M#?K]?

Strings: [ROTATE, 3, \n, \n, IF, BLOCKEDP, [, MOVE, 1, \n, NOP, ], \n]

Parser result: accepted

-----

Enter one or more lines. End with a single line with \$

If you want to stop testing, enter EXIT

ROTATE 3

IF BLOCKEDP [MOVE 1  
NOP]

(BLOCK  
IF BLOCKEDP [MOVE 1  
NOP]  
LEFT 90  
)  
\$

The input was accepted by the lexer.

Token Stream: R#???w[M#?K]??(b??w[M#?K]?1#?)?

Strings: [ROTATE, 3,\n,\n,IF, BLOCKEDP,[,MOVE, 1,\n,NOP,],\n,\n,(,BLOCK,\n,IF, BLOCKEDP,[,MOVE, 1,\n,NOP,],\n,LEFT, 90,\n,),\n]

Parser result: accepted

-----

Enter one or more lines. End with a single line with \$

If you want to stop testing, enter EXIT

```
ROTATE 3
```

```
IF BLOCKEDP [MOVE 1  
NOP]
```

```
(BLOCK  
IF BLOCKEDP [MOVE 1  
NOP]  
LEFT 90  
)
```

```
DEFINE one 1
```

```
TO foo :c :p  
OUTPUT  
DROP :c  
FREE :p  
MOVE one  
END
```

```
foo 1 3
```

```
$
```

The input was accepted by the lexer.

Token Stream: R#???w[M#?K]??(b???w[M#?K]?l#?)??dv#??Tvxx?O?Dx?Fx?Mv?\$??v##?

Strings: [ROTATE, 3,\n,\n,IF, BLOCKEDP,[,MOVE, 1,\n,NOP,],\n,\n,(,BLOCK,\n,IF, BLOCKEDP,[,MOVE, 1,\n,NOP,],\n,LEFT, 90,\n,),\n,\n,DEFINE, one

Parser result: accepted

# EJEMPLO ENUNCIADO

```
-----
Enter one or more lines. End with a single line with $
If you want to stop testing, enter EXIT
ROTATE 3

IF BLOCKEDP [MOVE 1
NOP]

(BLOCK
IF BLOCKEDP [MOVE 1
NOP]
LEFT 90
)

DEFINE one 1

TO foo :c :p
OUTPUT
DROP :c
FREE :p
MOVE one
END

foo 1 3
TO goend
OUTPUT IF !BLOCKEDP [
(BLOCK MOVE 1
goend)
NOP
]
END
$
The input was accepted by the lexer.
Token Stream: R#???w[M#?K]??(b???w[M#?K]?l#?)??dv#??Tvxx?O?Dx?Fx?Mv?$$??v##??Tv?O?!w[?(bM#?v)?K?]?$$
Strings: [ROTATE, 3,\n,\n,IF, BLOCKEDP,[,MOVE, 1,\n,NOP,],\n,\n,(,BLOCK,\n,IF, BLOCKEDP,[,MOVE, 1,\n,NOP,],\n,LEFT, 90,\n,),\n,\n,DEFINE, one
Parser result: accepted
-----
```

# REJECTED

```
-----  
Enter one or more lines. End with a single line with $  
If you want to stop testing, enter EXIT  
REPEAT 3 [  
MOVE 1  
LEFT 80  
FREE 5]  
$  
The input was  accepted by the lexer.  
Token Stream: X#[?M#?l#?F#]?  
Strings: [REPEAT, 3,[,\n,MOVE, 1,\n,LEFT, 80,\n,FREE, 5,],\n]  
Parser result: rejected  
-----
```

# ACCEPTED

```
-----  
Enter one or more lines. End with a single line with $  
If you want to stop testing, enter EXIT  
REPEAT 3 [  
MOVE 1  
LEFT 80  
FREE 5  
]  
$  
The input was  accepted by the lexer.  
Token Stream: X#[?M#?l#?F#]?  
Strings: [REPEAT, 3,[,\n,MOVE, 1,\n,LEFT, 80,\n,FREE, 5,\n,],\n]  
Parser result: accepted  
-----
```

# CHECK

```
-----  
Enter one or more lines. End with a single line with $  
If you want to stop testing, enter EXIT  
CHECK C 5  
$  
The input was  accepted by the lexer.  
Token Stream:  cC#?  
Strings: [CHECK, C, 5,\n]  
Parser result: accepted  
-----
```

```
-----  
Enter one or more lines. End with a single line with $  
If you want to stop testing, enter EXIT  
CHECK B 30  
$  
The input was  accepted by the lexer.  
Token Stream:  cB#?  
Strings: [CHECK, B, 30,\n]  
Parser result: accepted  
-----
```

# LOOK

-----  
Enter one or more lines. End with a single line with \$  
If you want to stop testing, enter EXIT

LOOK N

\$

The input was accepted by the lexer.

Token Stream: LN?

Strings: [LOOK, N,\n]

Parser result: accepted  
-----