

```

Enter the input string: a*(a/a)$
Input: a*(a/a)$
Stack: ['$ ', 'Q', 'T', '$'] Production Rule: E -> TQ
Input: a*(a/a)$
Stack: ['$ ', 'Q', 'R', 'F', '$'] Production Rule: T -> FR
Input: a*(a/a)$
Stack: ['$ ', 'Q', 'R', 'a', '$'] Production Rule: F -> a
Input: *(a/a)$
Stack: ['$ ', 'Q', 'R', '$'] Production Rule: match a
Input: *(a/a)$
Stack: ['$ ', 'Q', 'R', 'F', '*', '$'] Production Rule: R -> *FR
Input: (a/a)$
Stack: ['$ ', 'Q', 'R', 'F', '$'] Production Rule: match *
Input: (a/a)$
Stack: ['$ ', 'Q', 'R', ')', 'E', '(', '$'] Production Rule: F -> (E)
Input: a/a)$
Stack: ['$ ', 'Q', 'R', ')', 'E', '$'] Production Rule: match (
Input: a/a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'T', '$'] Production Rule: E -> TQ
Input: a/a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'F', '$'] Production Rule: T -> FR
Input: a/a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'a', '$'] Production Rule: F -> a
Input: /a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', '$'] Production Rule: match a
Input: /a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'F', '/', '$'] Production Rule: R -> /FR
Input: a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'F', '$'] Production Rule: match /
Input: a)$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'a', '$'] Production Rule: F -> a
Input: )$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', '$'] Production Rule: match a
Input: )$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'ε', '$'] Production Rule: R -> ε
Input: )$
Stack: ['$ ', 'Q', 'R', ')', 'ε', '$'] Production Rule: Q -> ε
Input: $$
Stack: ['$ ', 'Q', 'R', '$'] Production Rule: match )
Input: $$
Stack: ['$ ', 'Q', 'ε', '$'] Production Rule: R -> ε
Input: $$
Stack: ['$ ', 'ε', '$'] Production Rule: Q -> ε
Input: $
Stack: ['$'] Production Rule: match $
String is accepted/ valid.

```

```

Enter the input string: (a+a)*a$
Input: (a+a)*a$$
Stack: ['$ ', 'Q', 'T', '$'] Production Rule: E -> TQ
Input: (a+a)*a$$
Stack: ['$ ', 'Q', 'R', 'F', '$'] Production Rule: T -> FR
Input: (a+a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'E', '(', '$'] Production Rule: F -> (E)
Input: a+a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'E', '$'] Production Rule: match (
Input: a+a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'T', '$'] Production Rule: E -> TQ
Input: a+a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'F', '$'] Production Rule: T -> FR
Input: a+a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'a', '$'] Production Rule: F -> a
Input: +a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', '$'] Production Rule: match a
Input: +a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'ε', '$'] Production Rule: R -> ε
Input: +a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'T', '+', '$'] Production Rule: Q -> +TQ
Input: a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'T', '$'] Production Rule: match +
Input: a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'F', '$'] Production Rule: T -> FR
Input: a)*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', 'a', '$'] Production Rule: F -> a
Input: )*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'R', '$'] Production Rule: match a
Input: )*a$$
Stack: ['$ ', 'Q', 'R', ')', 'Q', 'ε', '$'] Production Rule: R -> ε
Input: )*a$$
Stack: ['$ ', 'Q', 'R', ')', 'ε', '$'] Production Rule: Q -> ε
Input: *a$$
Stack: ['$ ', 'Q', 'R', '$'] Production Rule: match )
Input: *a$$
Stack: ['$ ', 'Q', 'R', 'F', '*', '$'] Production Rule: R -> *FR
Input: a$$
Stack: ['$ ', 'Q', 'R', 'F', '$'] Production Rule: match *
Input: a$$
Stack: ['$ ', 'Q', 'R', 'a', '$'] Production Rule: F -> a
Input: $$
Stack: ['$ ', 'Q', 'R', '$'] Production Rule: match a
Input: $$
Stack: ['$ ', 'Q', 'ε', '$'] Production Rule: R -> ε
Input: $$
Stack: ['$ ', 'ε', '$'] Production Rule: Q -> ε
Input: $
Stack: ['$'] Production Rule: match $
String is accepted/ valid.

```

```
Enter the input string: a(a+a)$
Input: a(a+a)$
Stack: ['$ ', 'Q', 'T', '$ '] Production Rule: E -> TQ
Input: a(a+a)$
Stack: ['$ ', 'Q', 'R', 'F', '$ '] Production Rule: T -> FR
Input: a(a+a)$
Stack: ['$ ', 'Q', 'R', 'a', '$ '] Production Rule: F -> a
Input: (a+a)$
Stack: ['$ ', 'Q', 'R', '$ '] Production Rule: match a
Input: (a+a)$
Stack: ['$ ', 'Q', '$ '] Production Rule: error
String is not accepted/ In valid.
```