Infix notation: 2 3 \* 4 2 - / 5 6 \* +



1. Setup walker, value stack, operand 1, Operand 2.
   1. The walker iterates the given string and evaluates it as a variable or character operator to perform the next operation.
   2. The value stack is a return value as a calculation result of infix notation.
   3. The operand 1 is a peek variable from the value stack.
   4. The operand 2 is another peek variable from the value stack.
2. The walker is “2”

It goes to the value stack since it is a variable.

1. The walker is “3”

It goes to the value stack since it is a variable.

1. The walker is “\*”
   1. Get peek (3) and set it to operand 2 and pop value stack.

Get peek (2) and set it to operand 1 and pop value stack.

* 1. Calculate operand 1 (2) and operand 1 (3) with walker (\*).

The result will be 6 as shown in the column.

It is pushed to the value stack.

1. The walker is “4”

It goes to the value stack since it is a variable.

1. The walker is “2”

It goes to the value stack since it is a variable.

1. The walker is “-”
   1. Get peek (2) and set it to operand 2 and pop value stack.

Get peek (4) and set it to operand 1 and pop value stack.

* 1. Calculate operand 1 (4) and operand 2 (2) with walker (-)

The result will be 2 as shown in the column.

It is pushed to the value stack.

1. The walker is “/”
   1. Get peek (2) and set it to operand 2 and pop value stack.

Get peek (6) and set it to operand 1 and pop value stack.

* 1. Calculate operand 1 (6) and operand 2 (2) with walker (/)

The result will be 3 as shown in the column.

It is pushed to the value stack.

1. The walker is “5”

It goes to the value stack since it is a variable.

1. The walker is “6”

It goes to the value stack since it is a variable.

1. The walker is “\*”
   1. Get peek (6) and set it to operand 2 and pop value stack.

Get peek (5) and set it to operand 1 and pop value stack.

* 1. Calculate operand 1 (5) and operand 1 (6) with walker (\*).

The result will be 30 as shown in the column.

It is pushed to the value stack.

1. The walker is “+”
   1. Get peek (30) and set it to operand 2 and pop value stack.

Get peek (3) and set it to operand 1 and pop value stack.

* 1. Calculate operand 1 (3) and operand 2 (30) with walker (+)

The result will be 33 as shown in the column.

It is pushed to the value stack.

1. Return value stack peek as a total calculation result.