

# Project 1 - A stack algorithm example

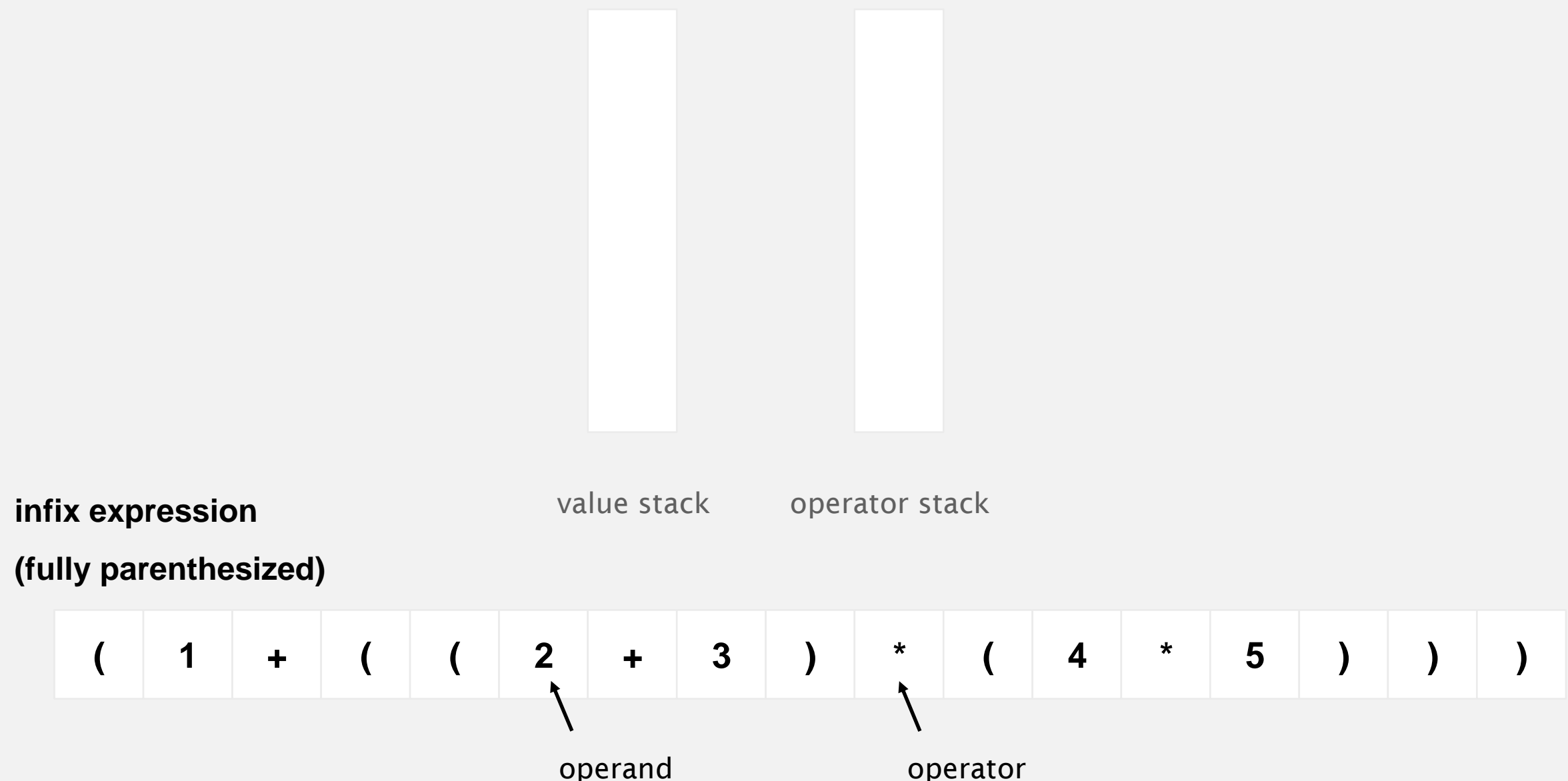
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



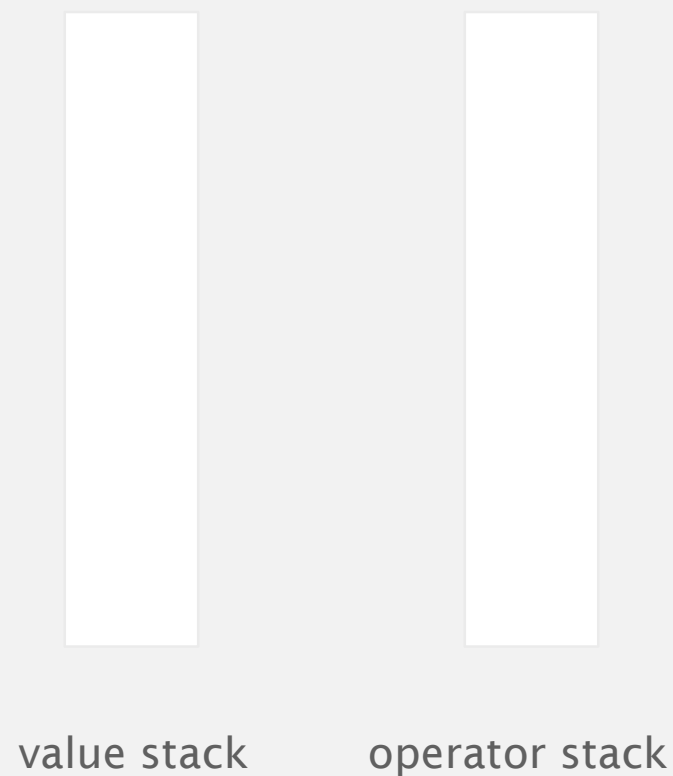
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



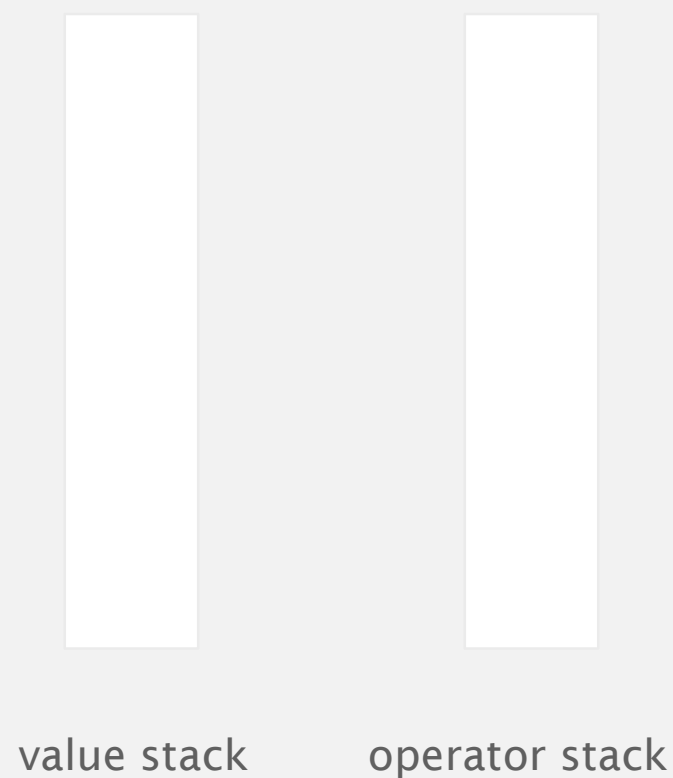
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



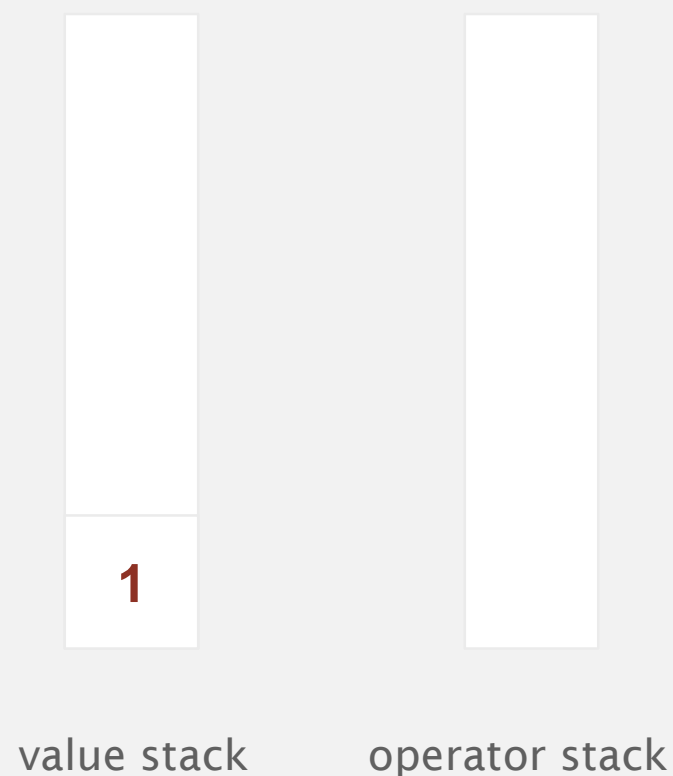
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



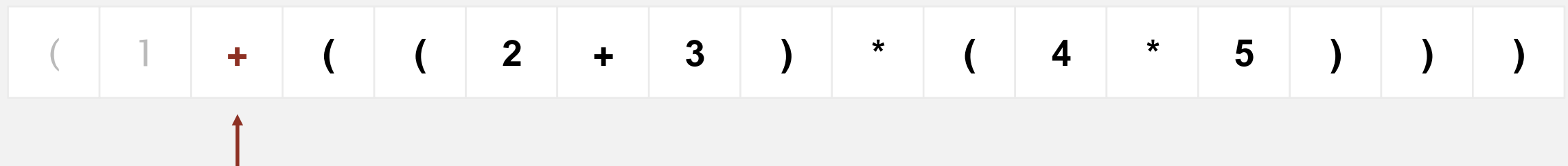
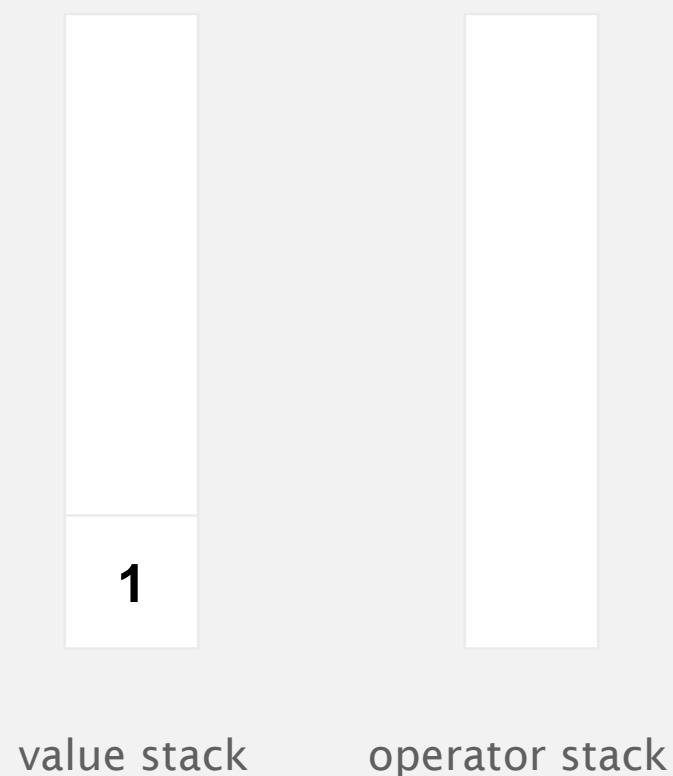
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



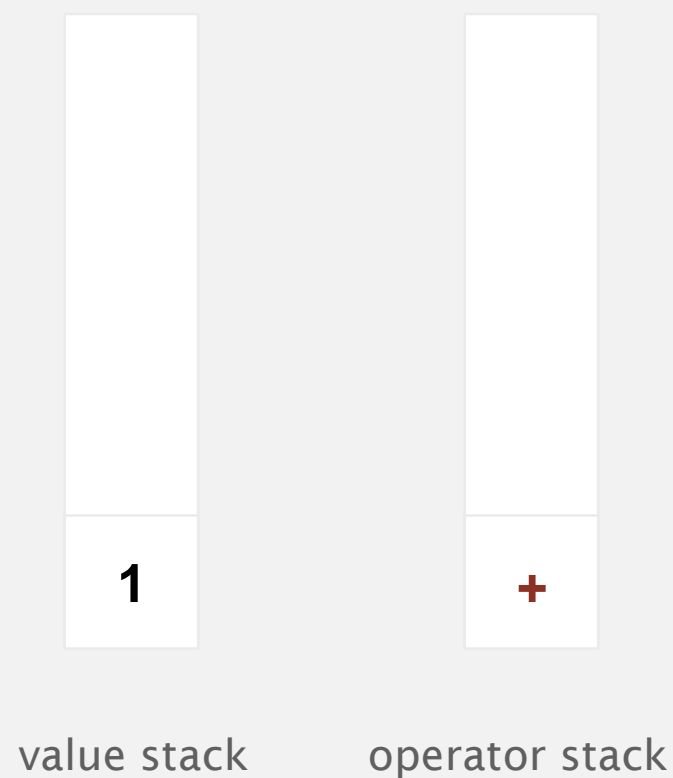
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



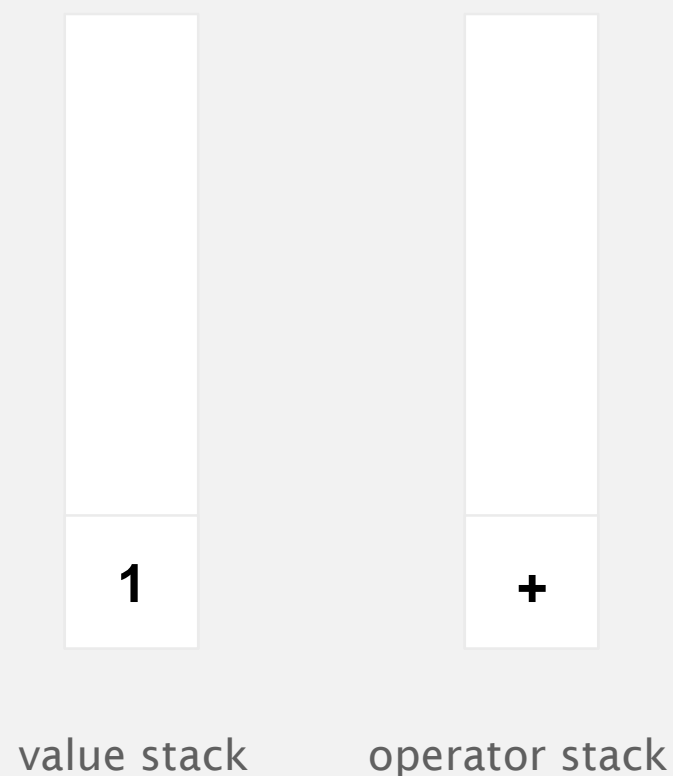
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



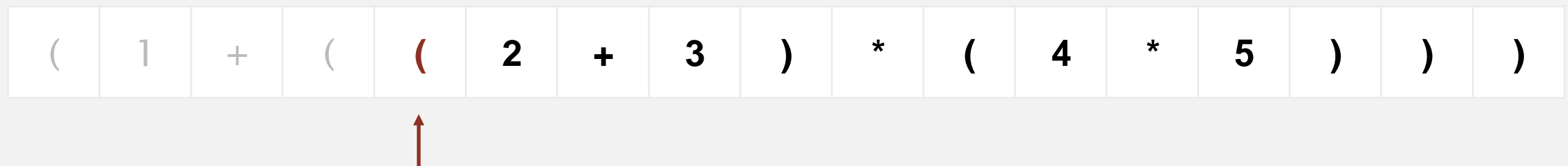
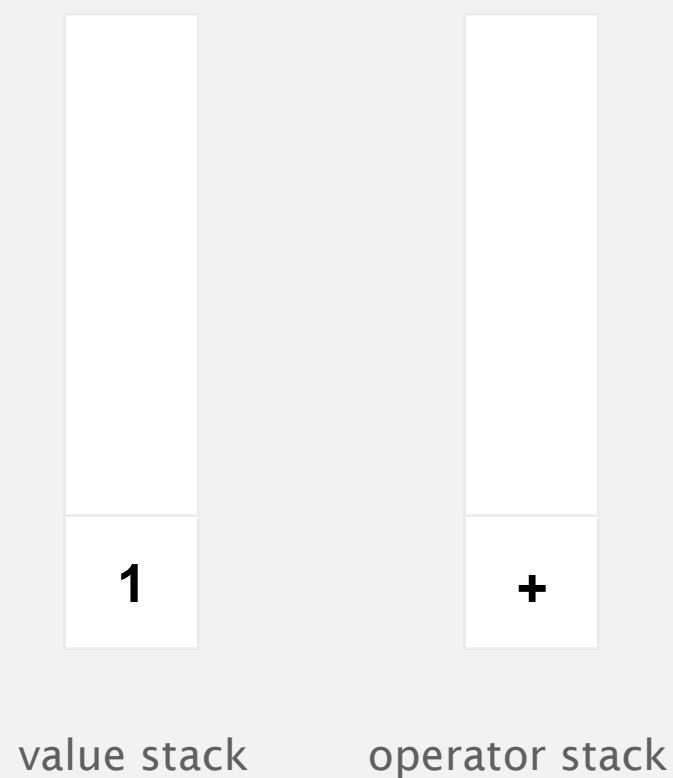
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.





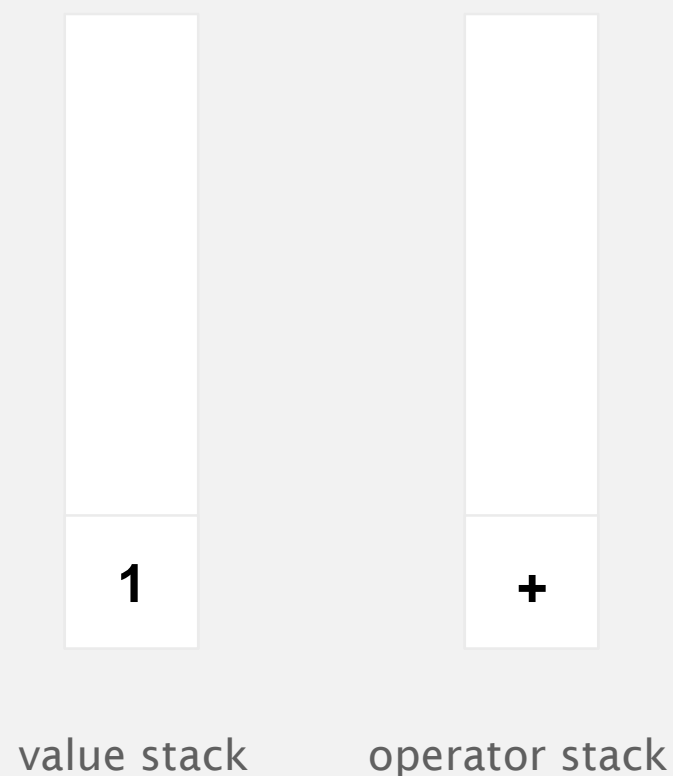
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



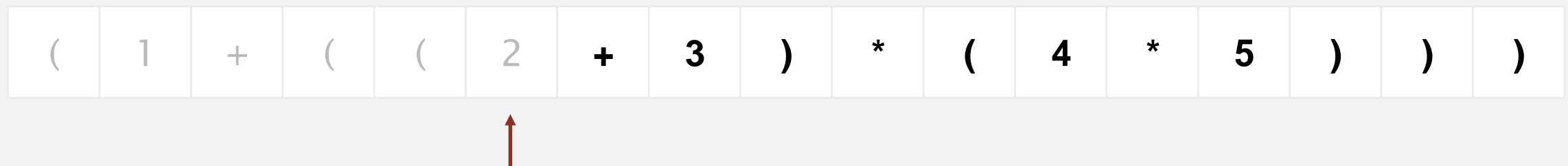
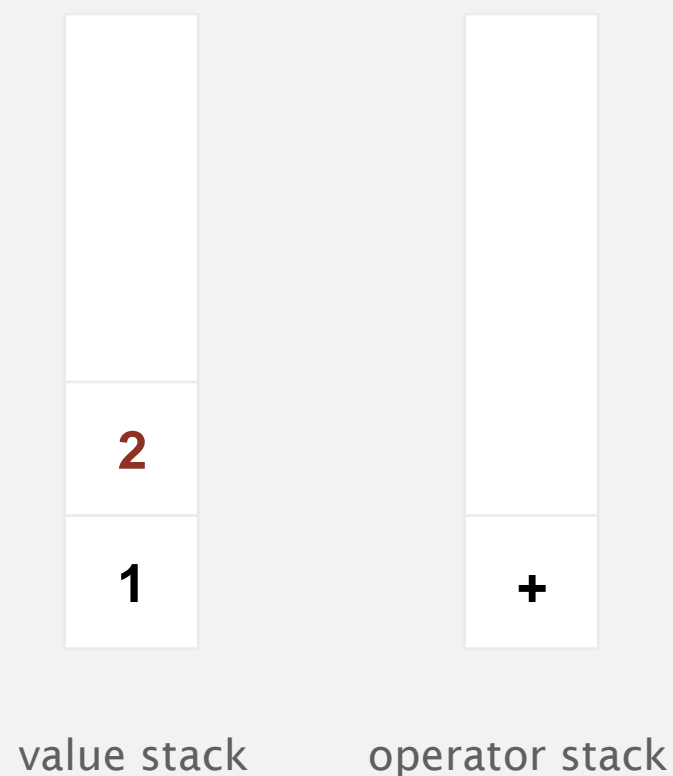
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



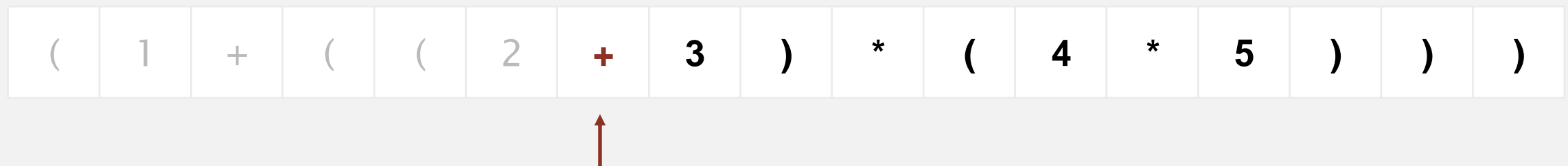
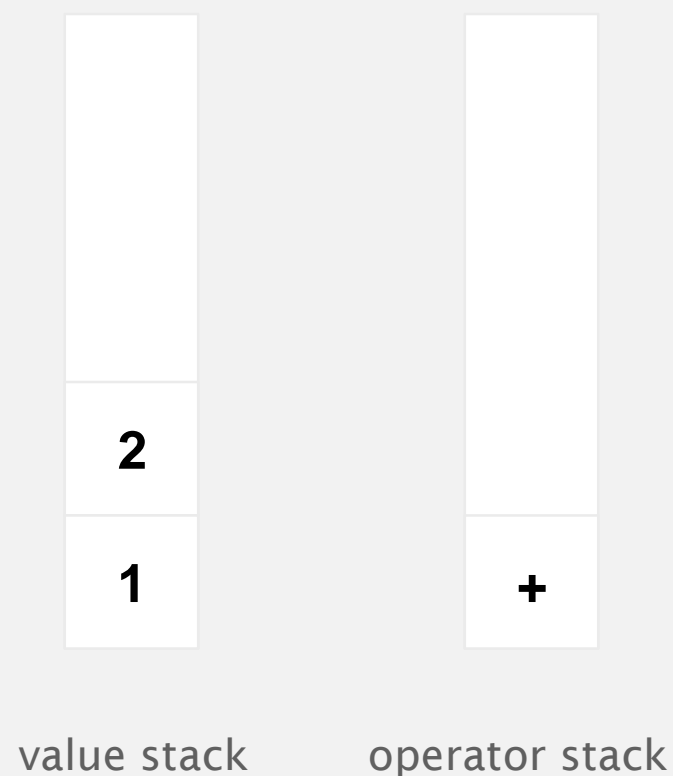
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



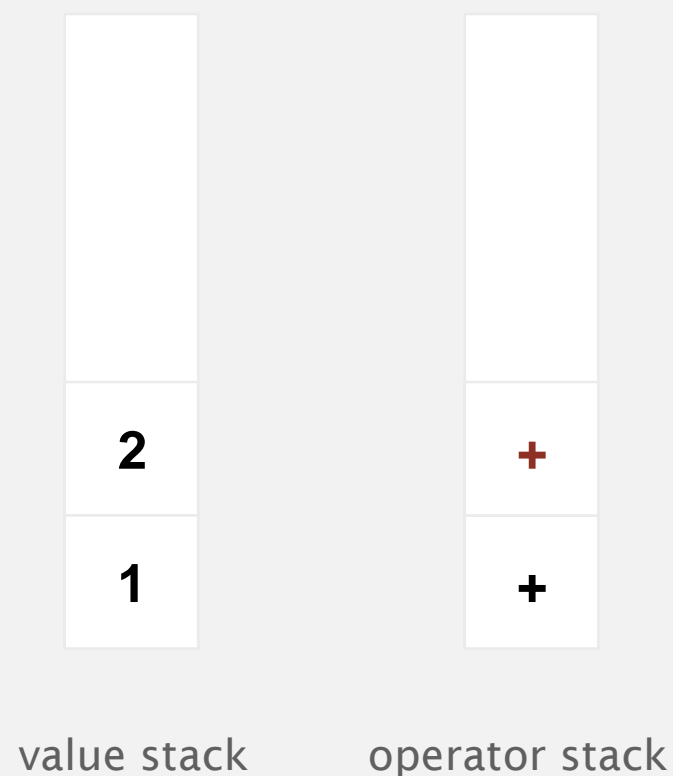
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



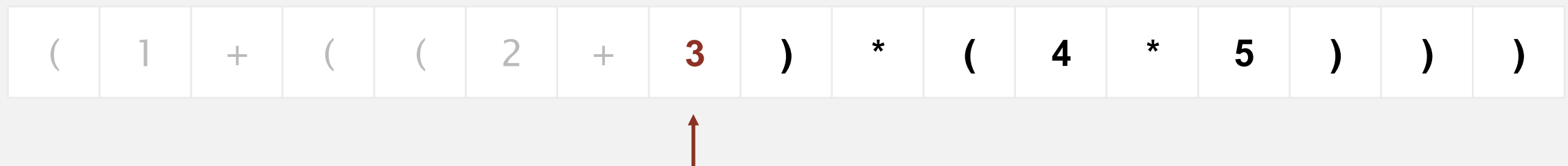
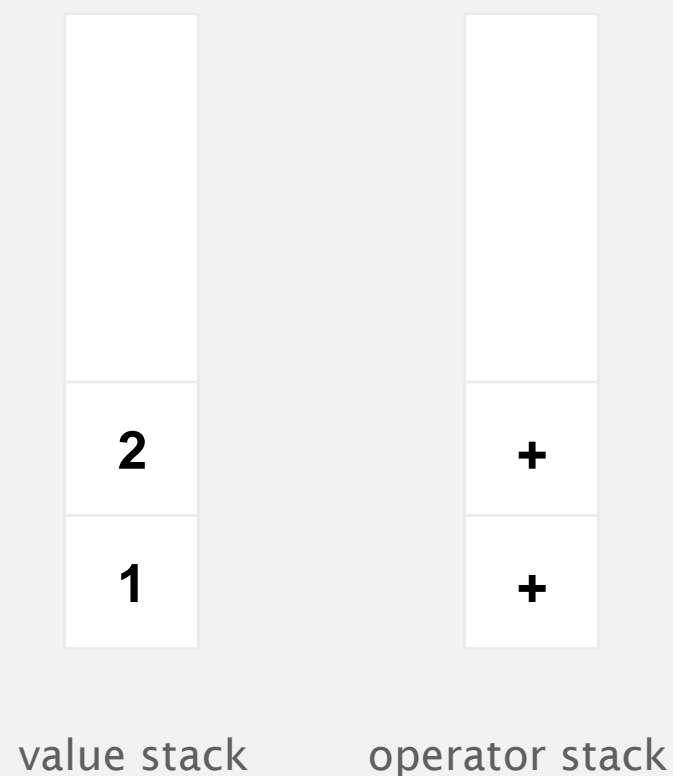
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



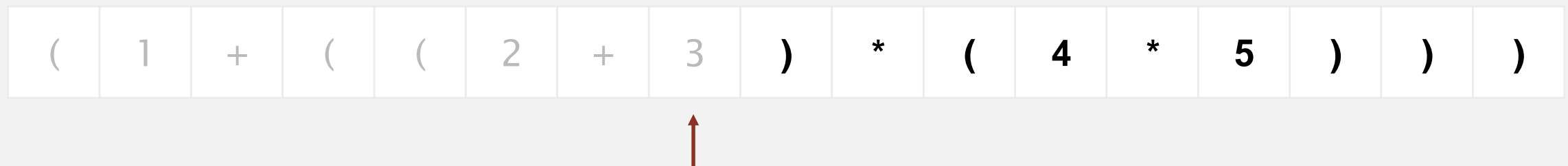
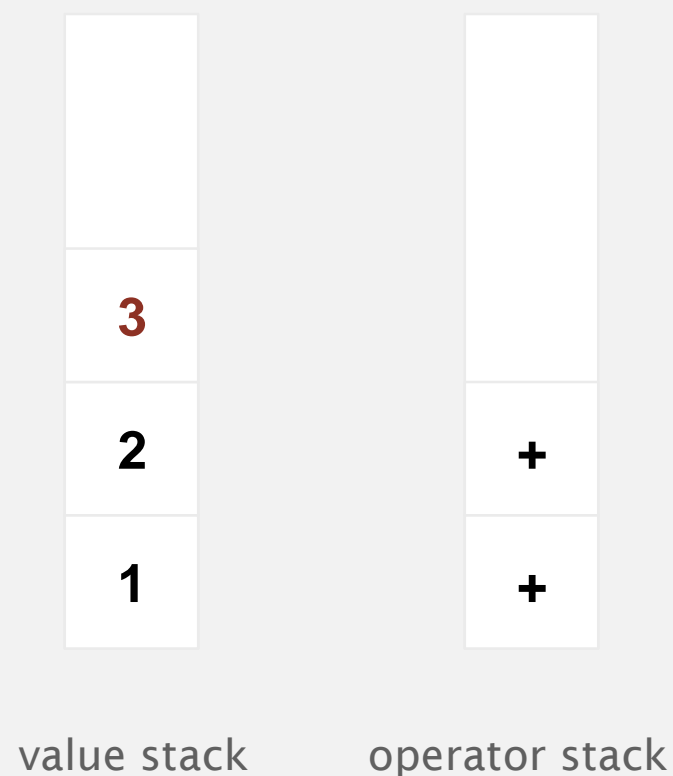
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



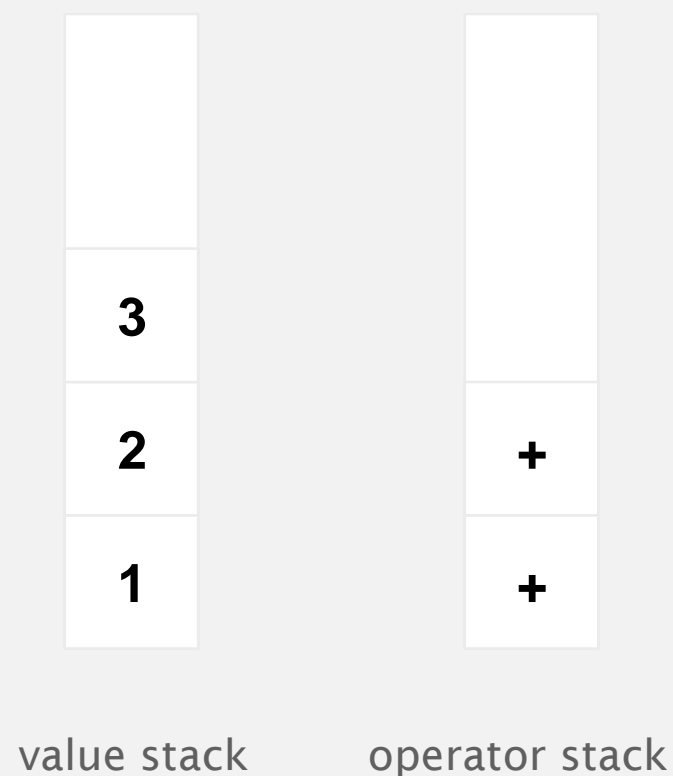
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



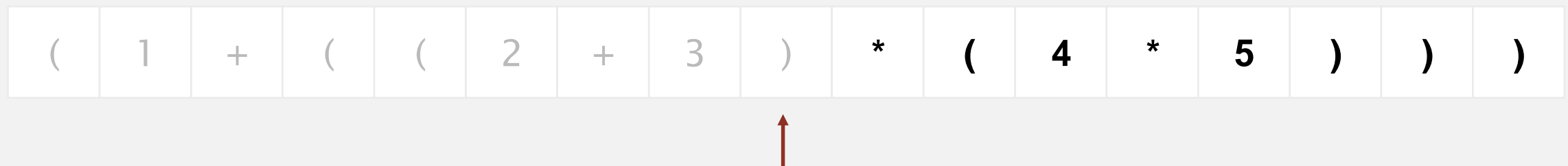
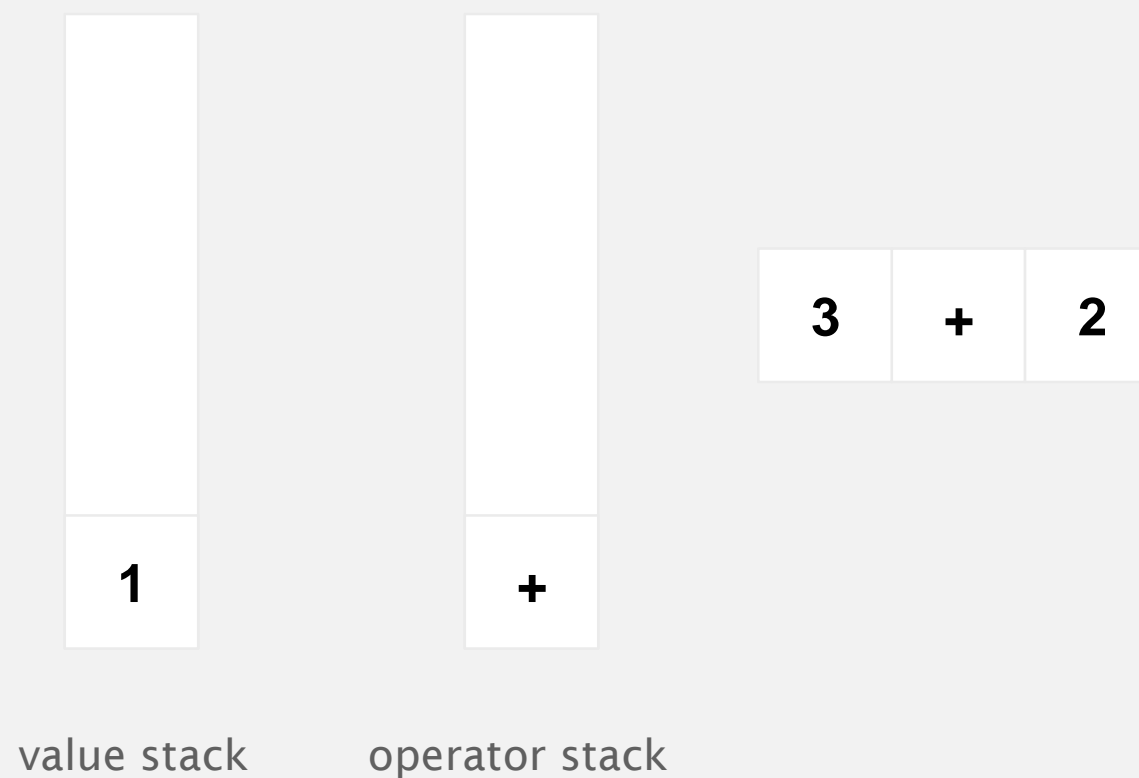
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.





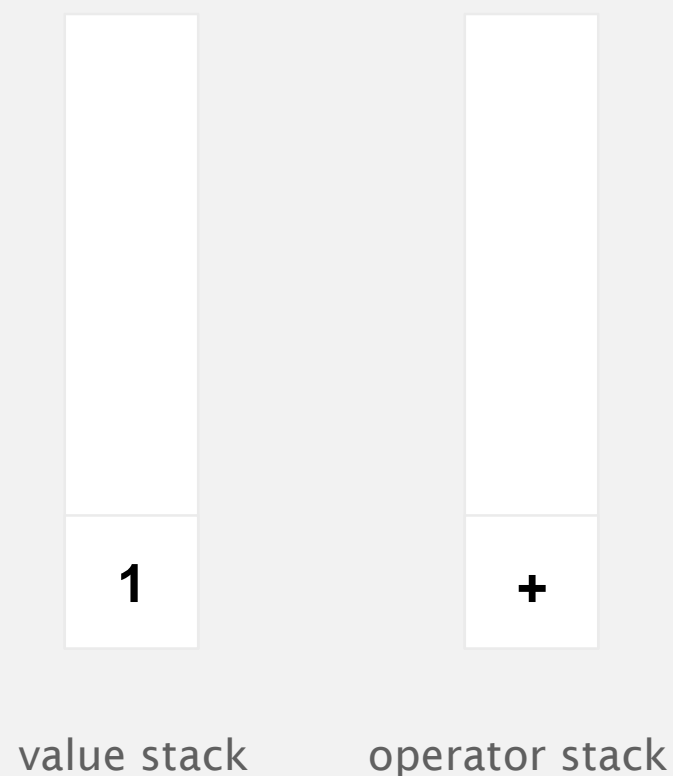
---

**Value:** push onto the value stack.

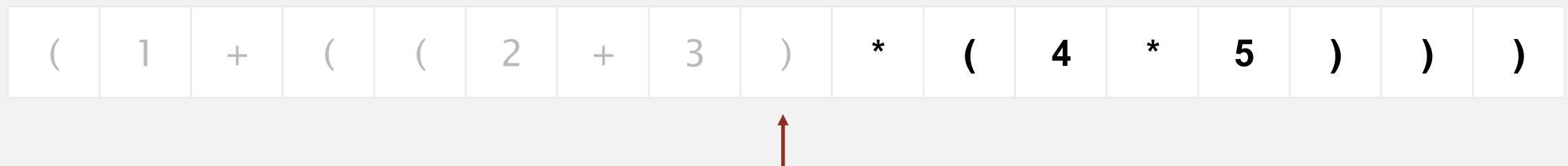
**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



3 + 2 = 5



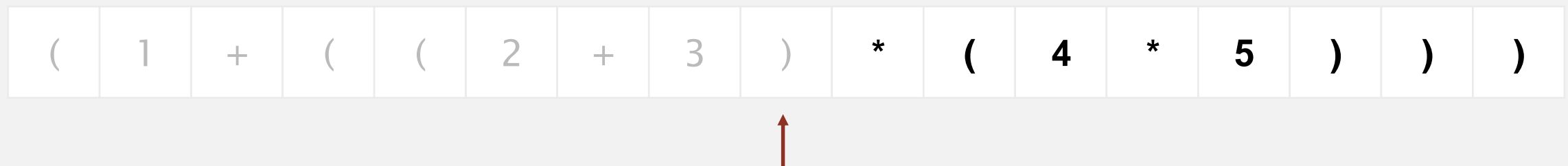
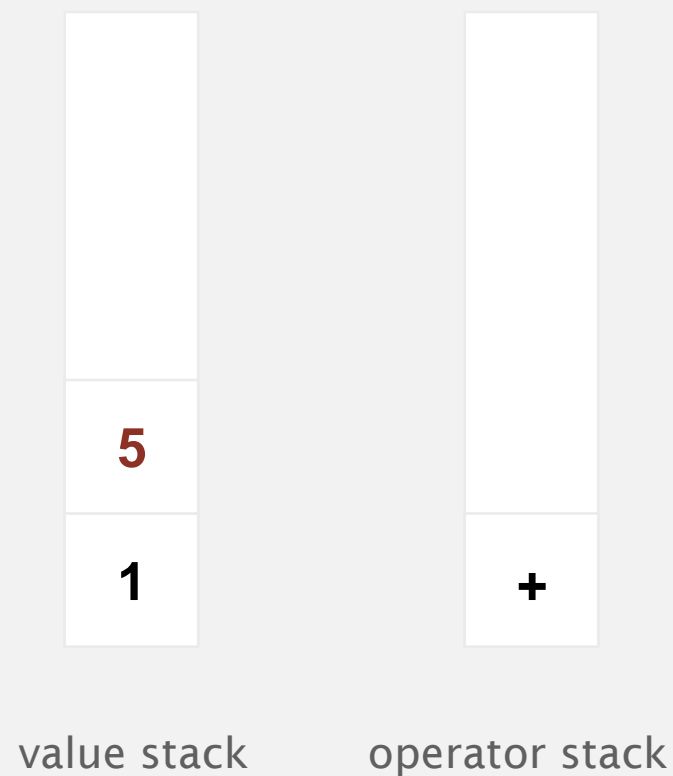
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



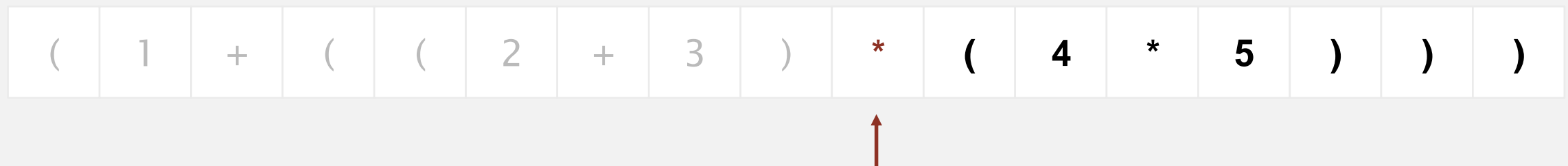
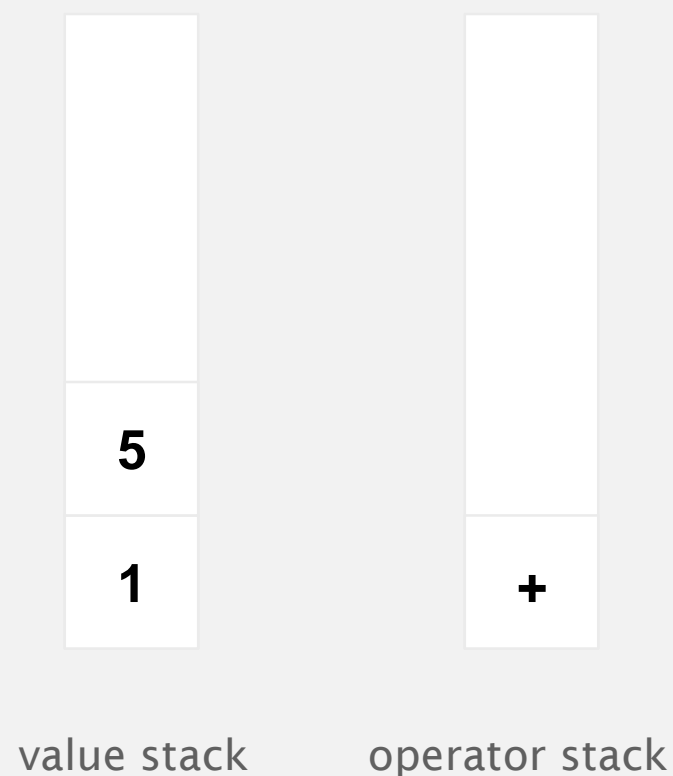
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



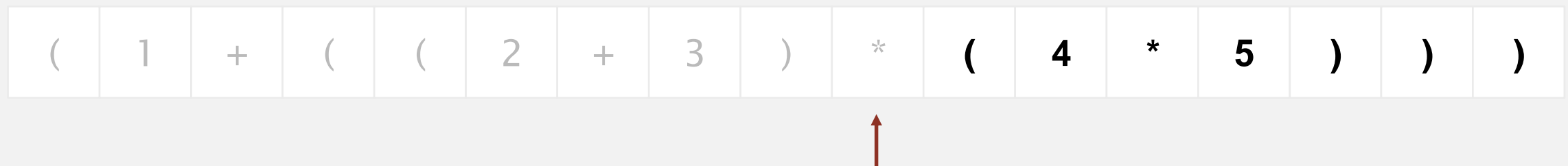
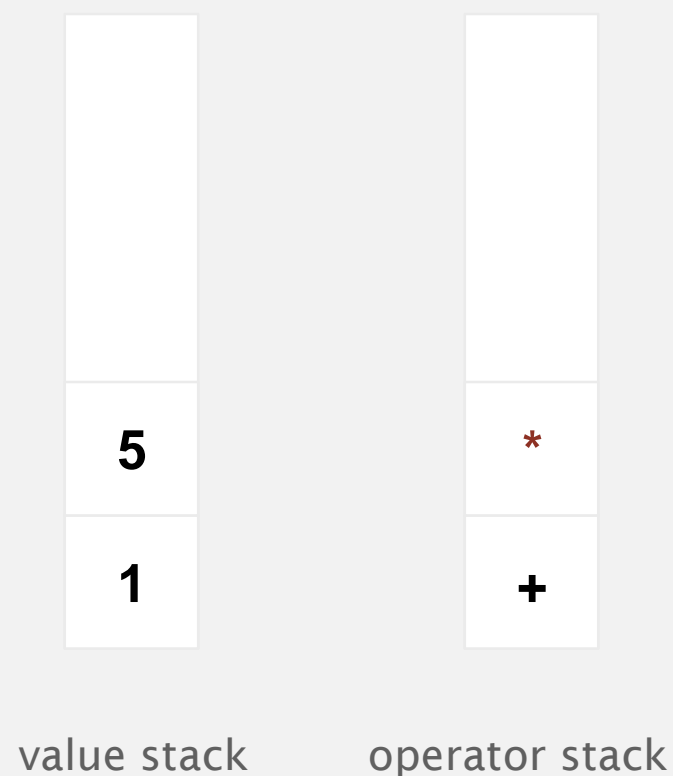
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



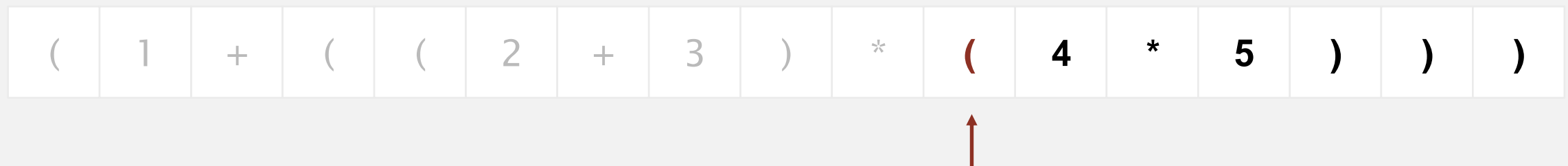
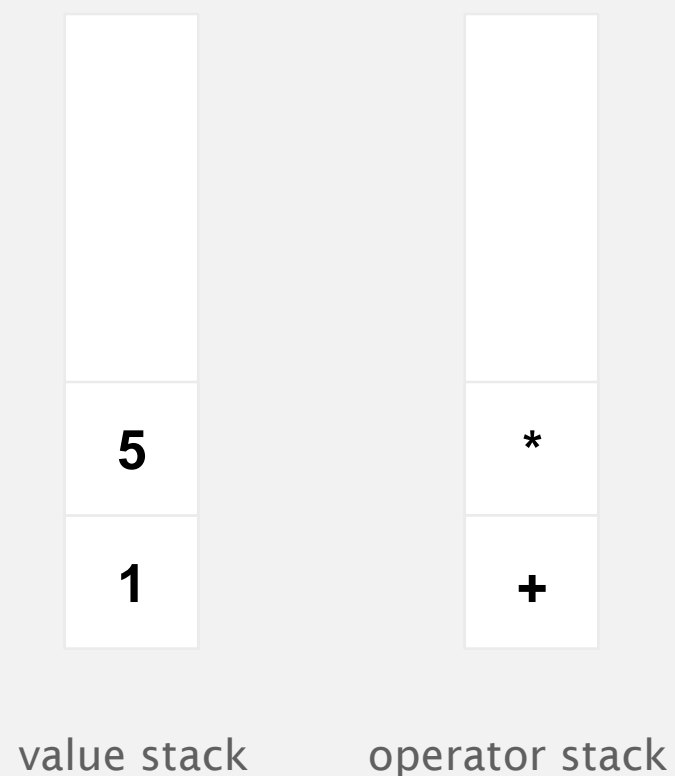
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



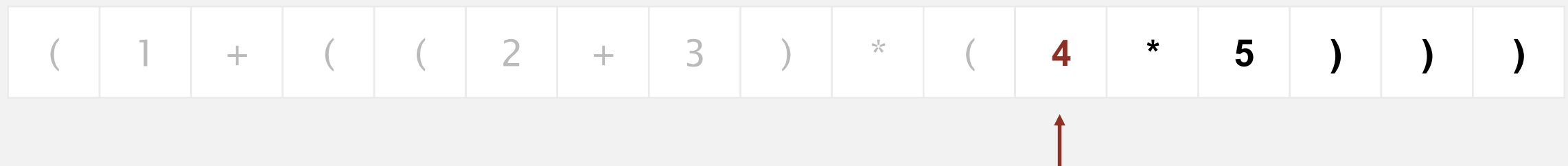
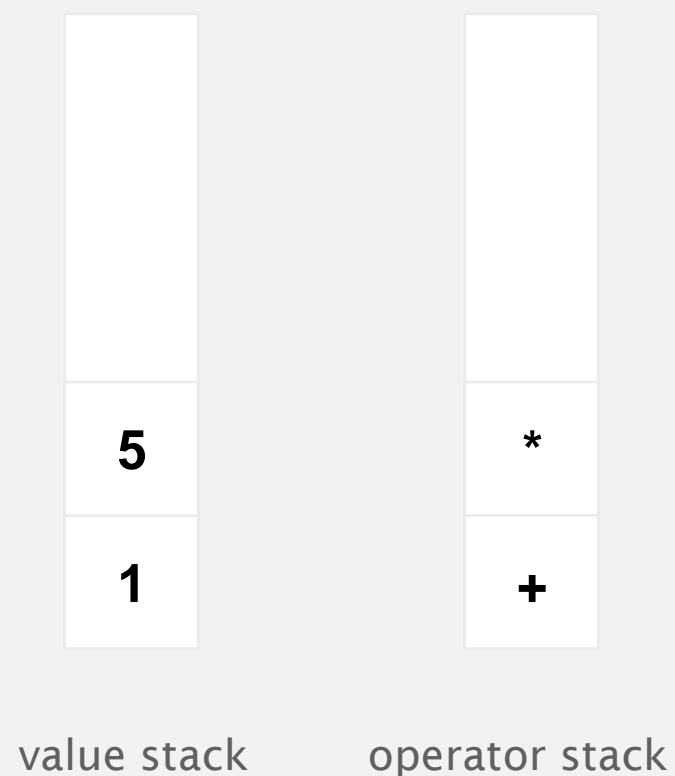
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



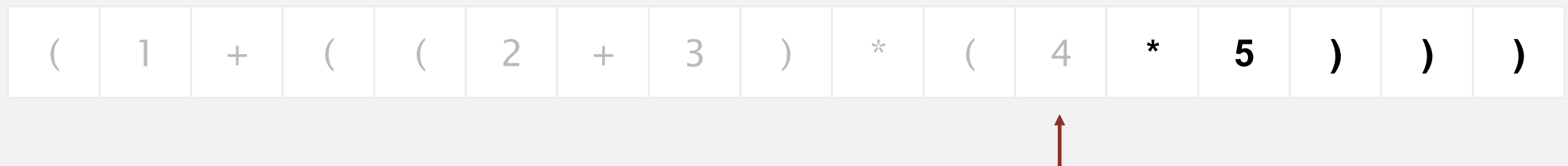
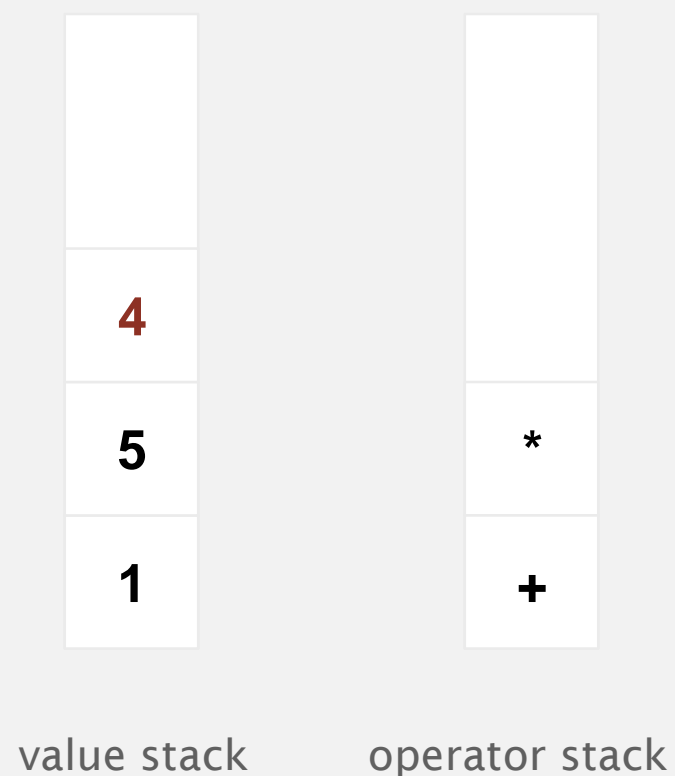
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



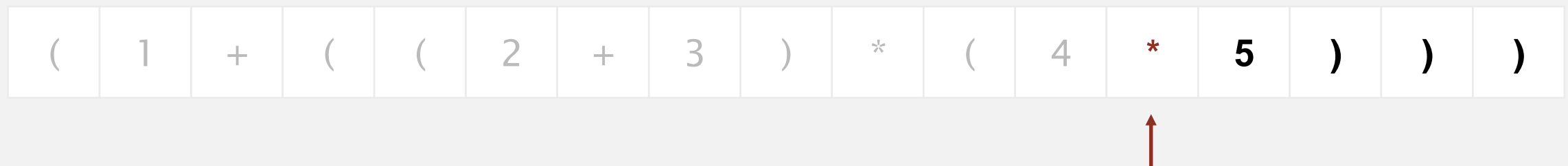
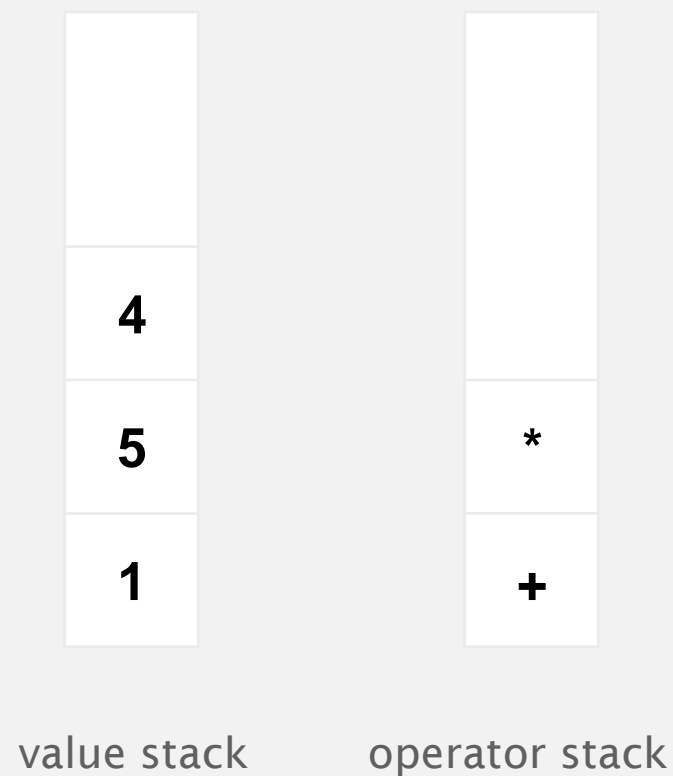
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.





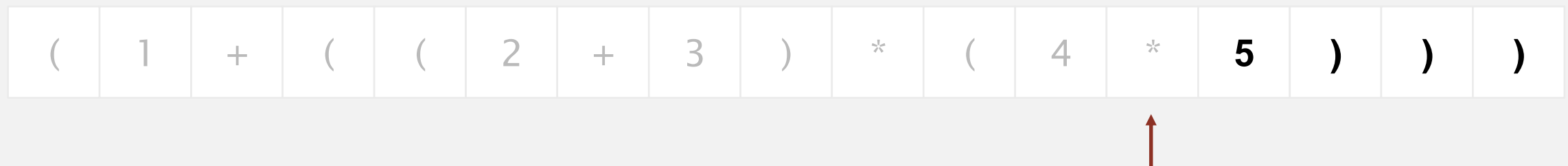
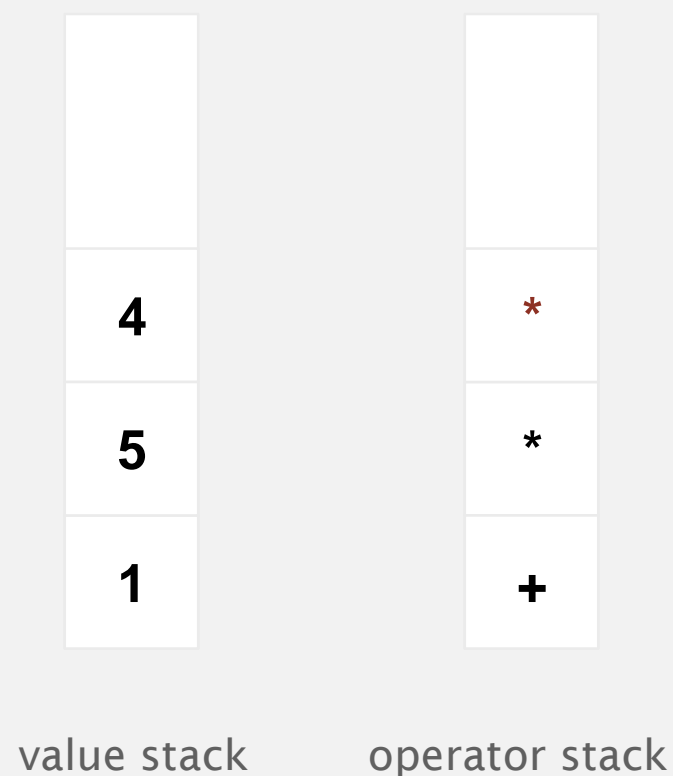
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



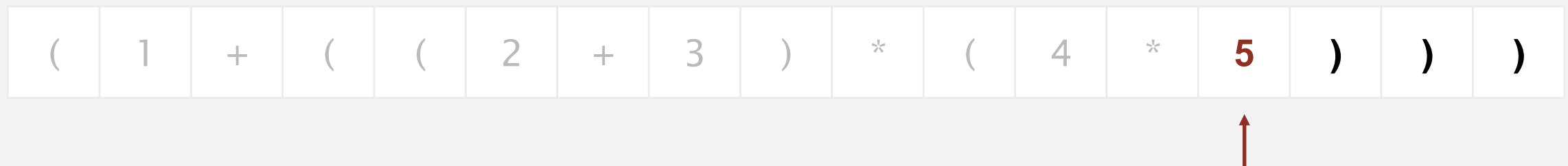
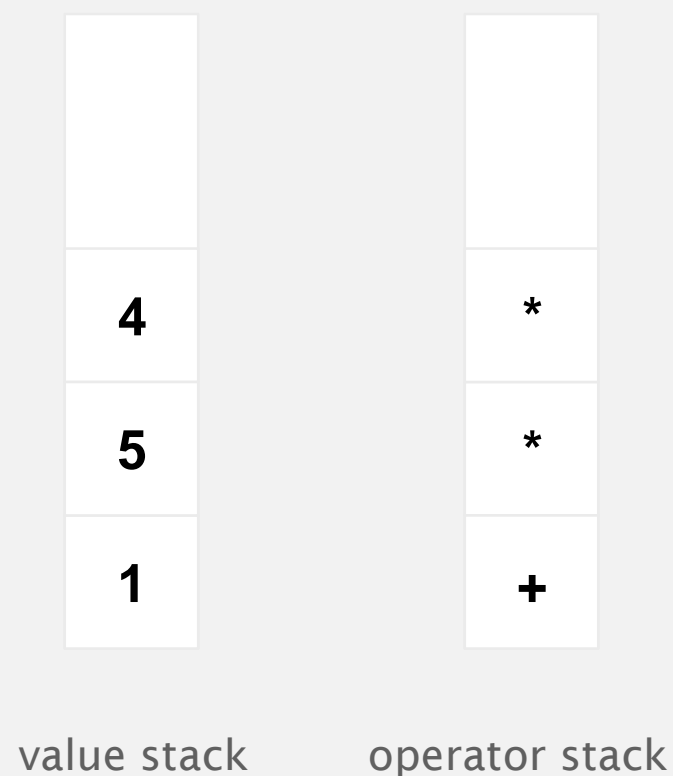
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



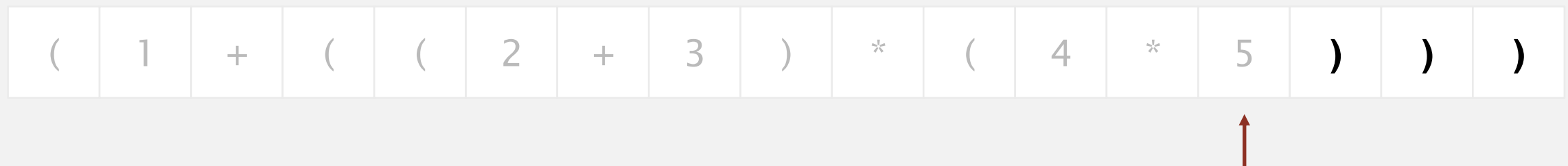
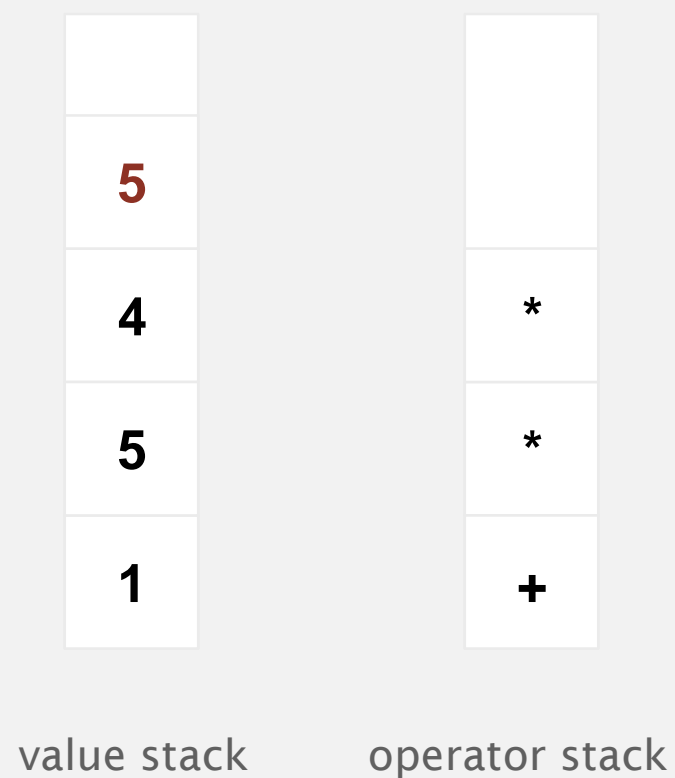
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



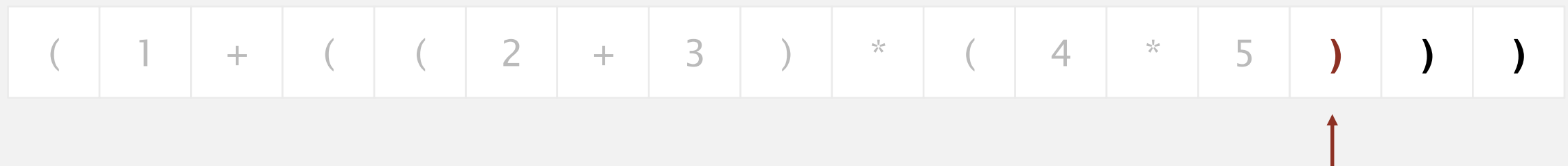
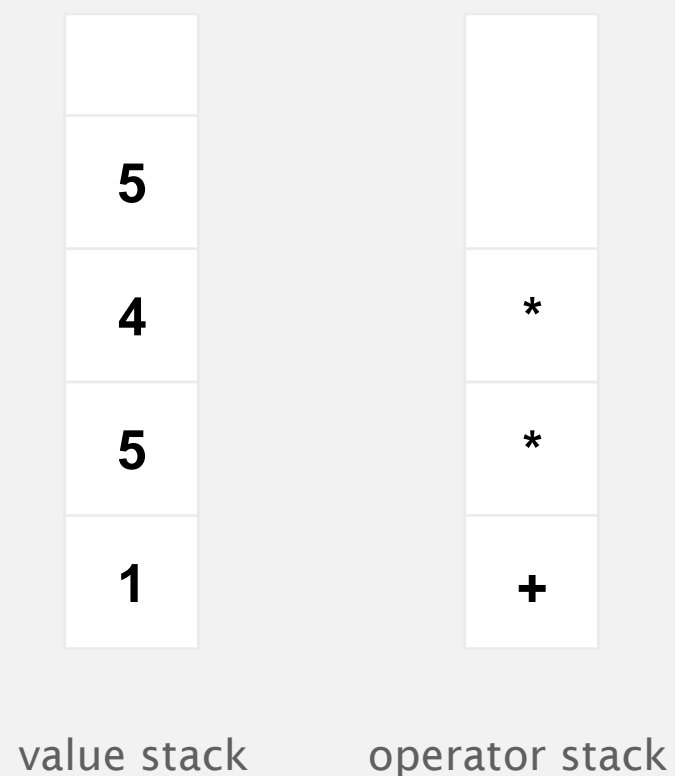
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



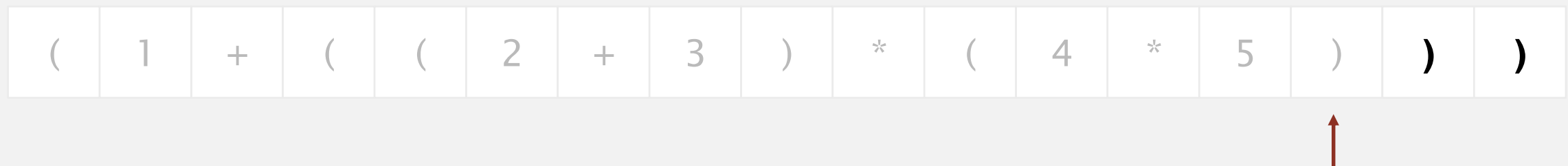
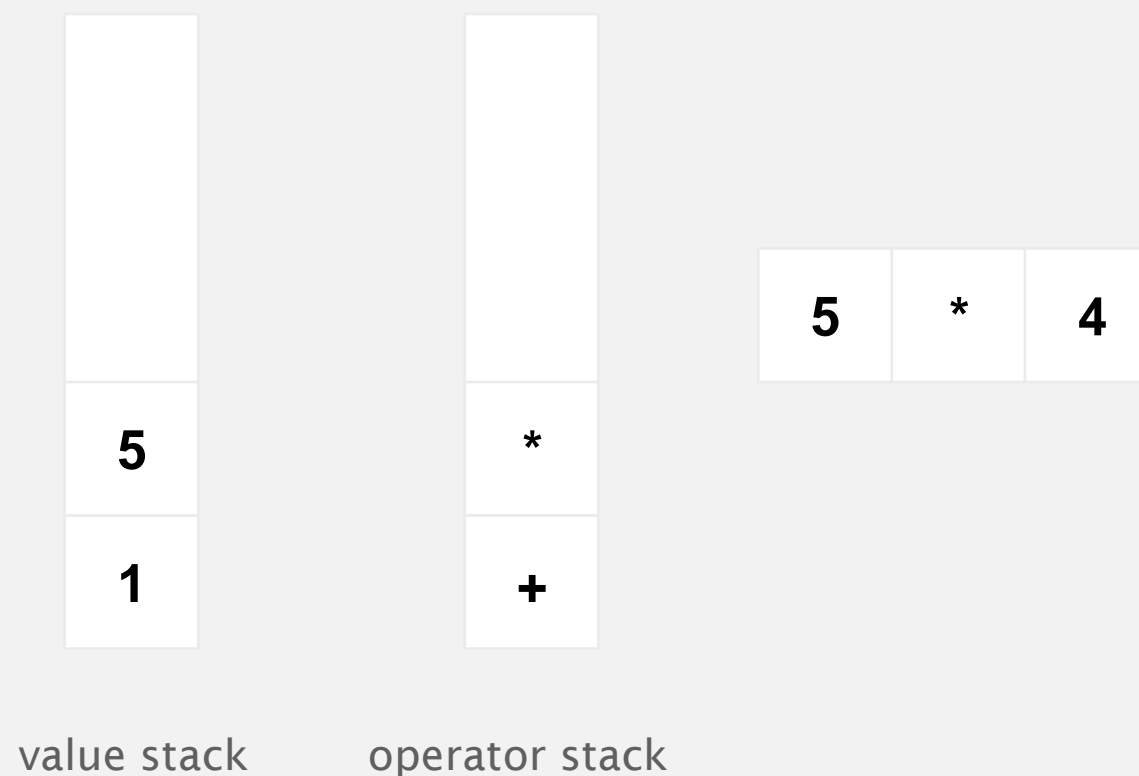
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



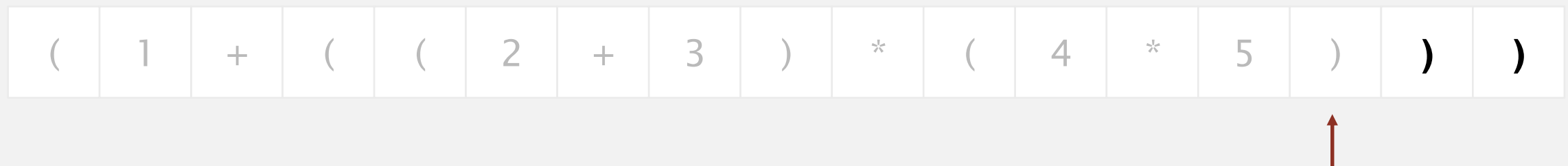
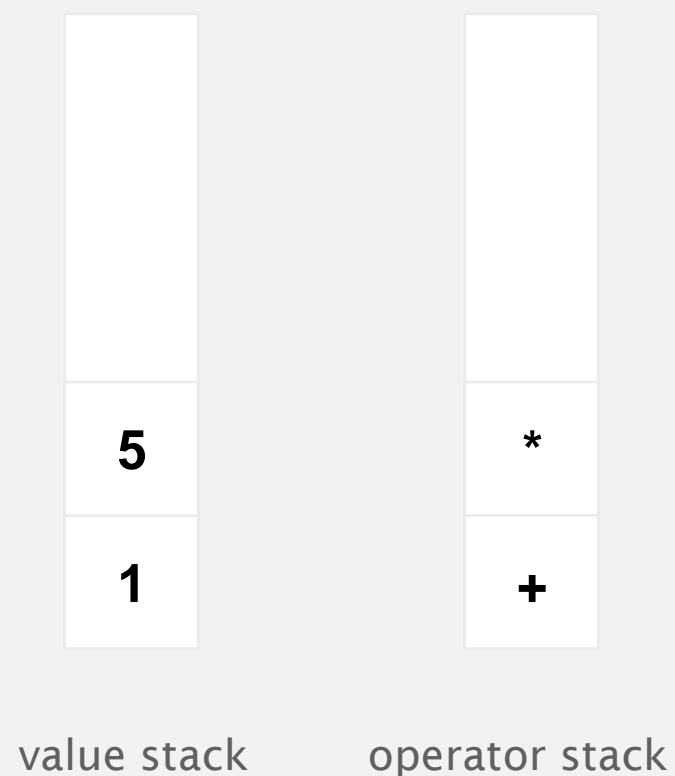
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



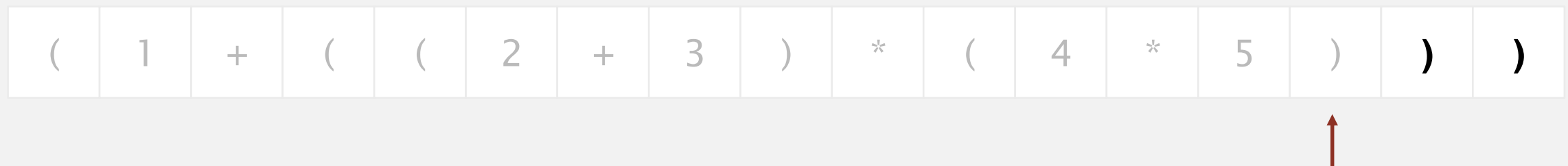
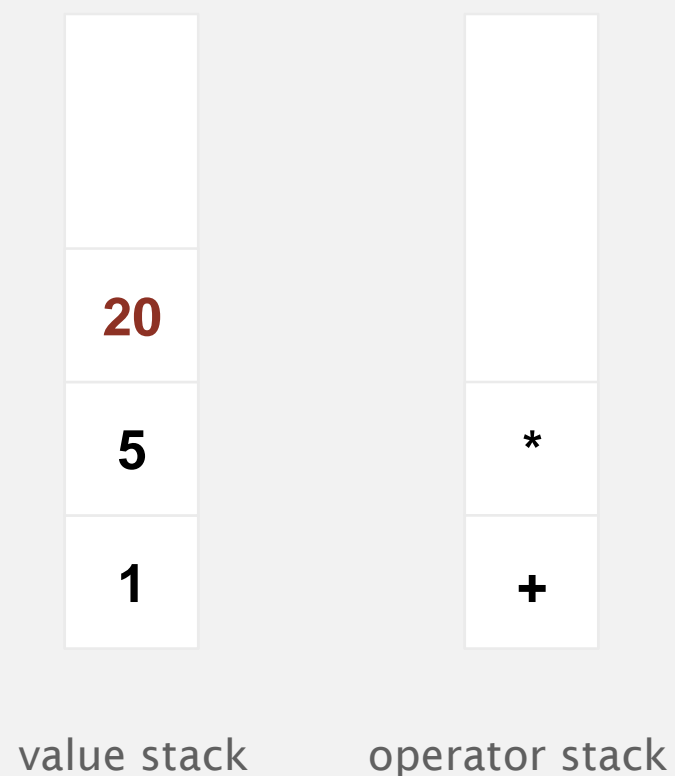
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



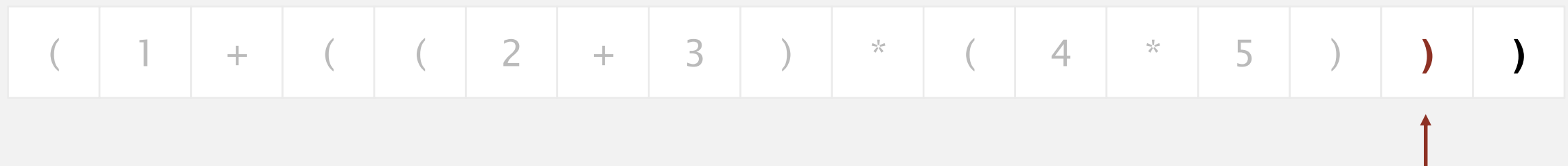
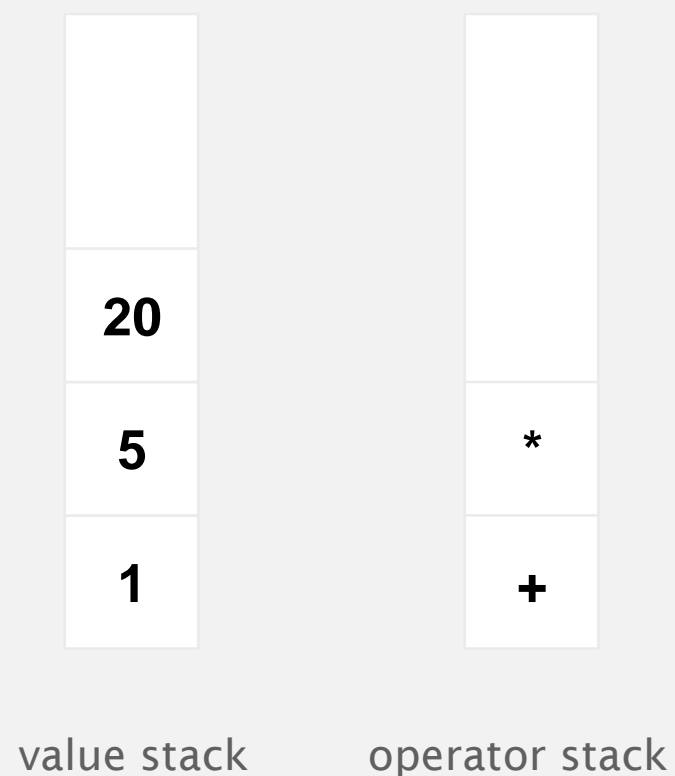
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.





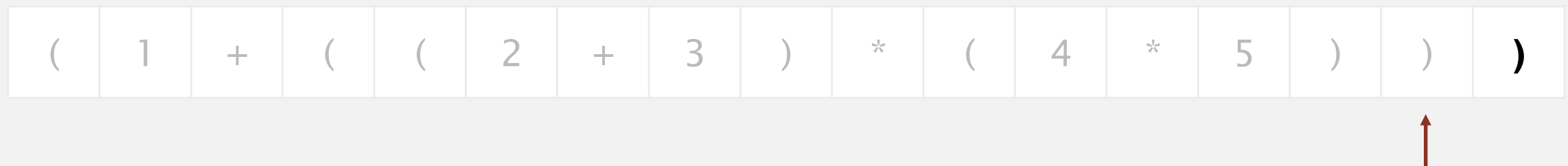
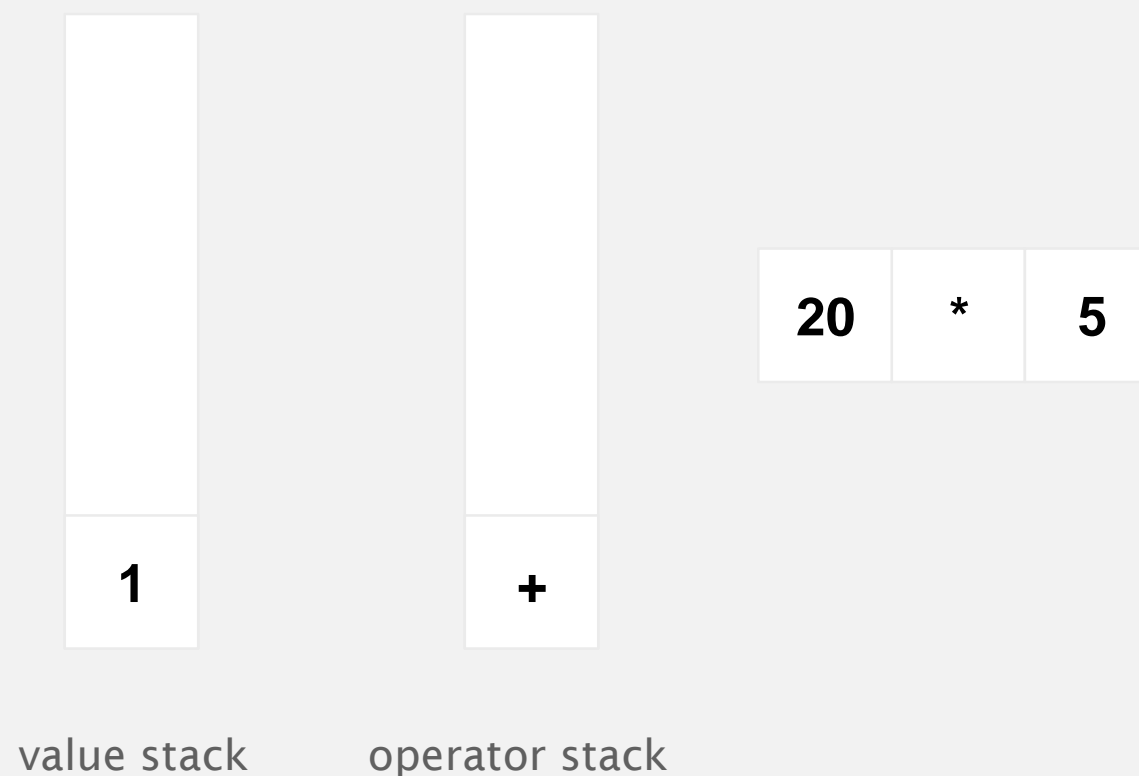
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



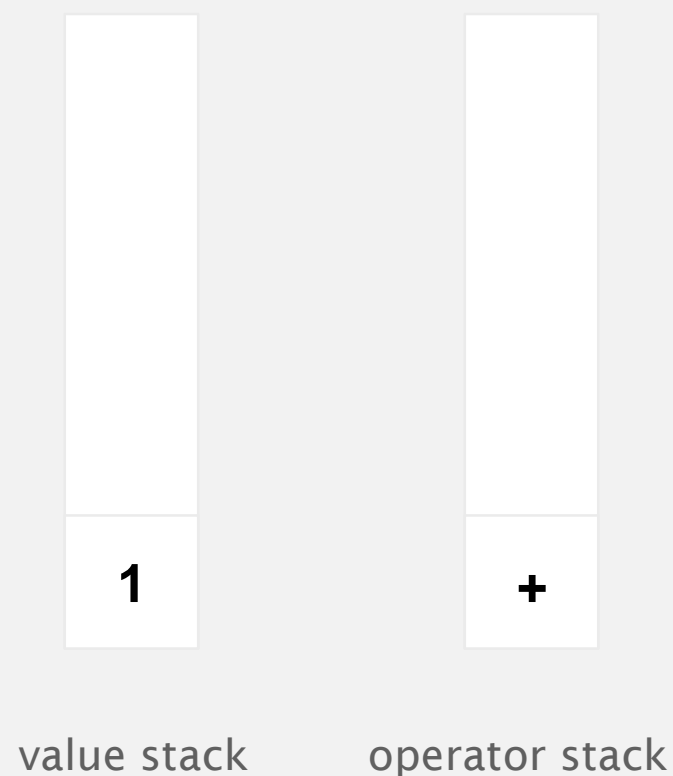
---

**Value:** push onto the value stack.

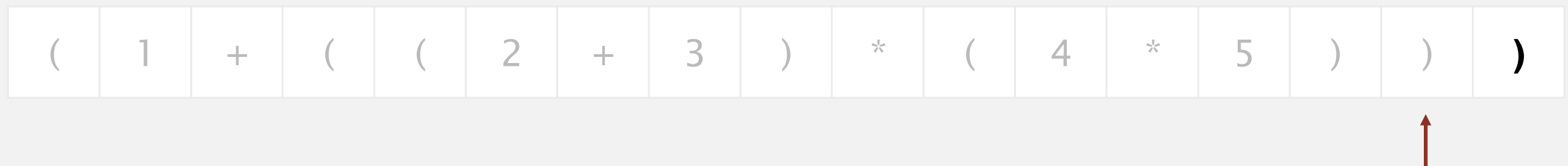
**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



20	*	5	=	100
----	---	---	---	-----



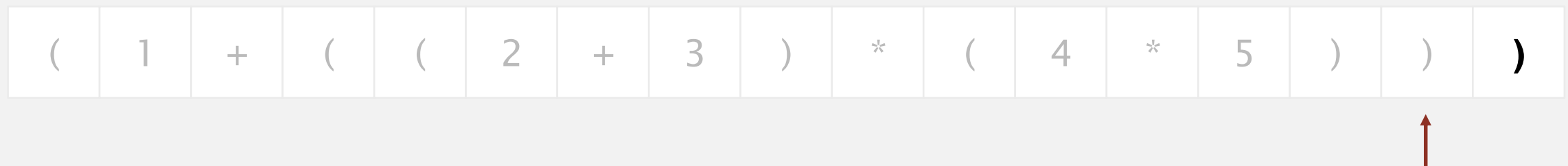
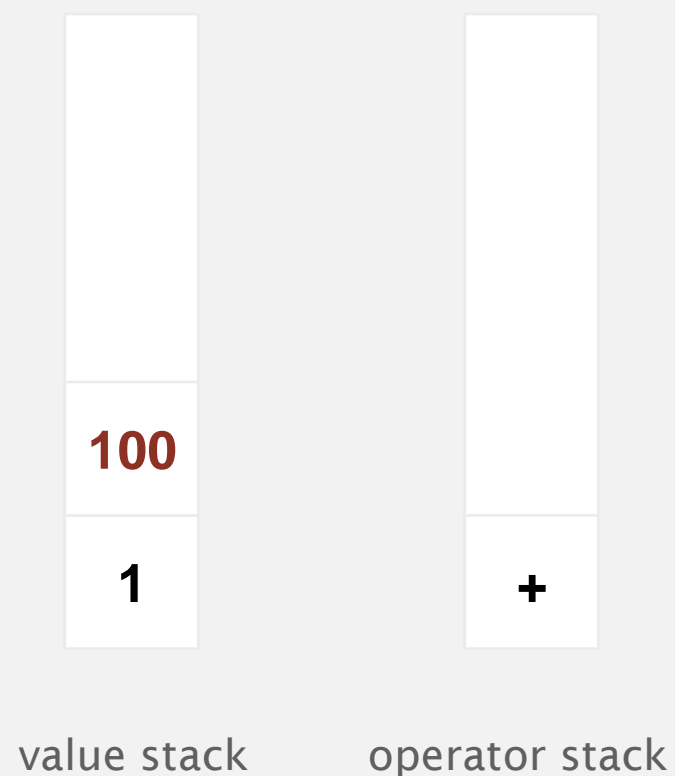
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



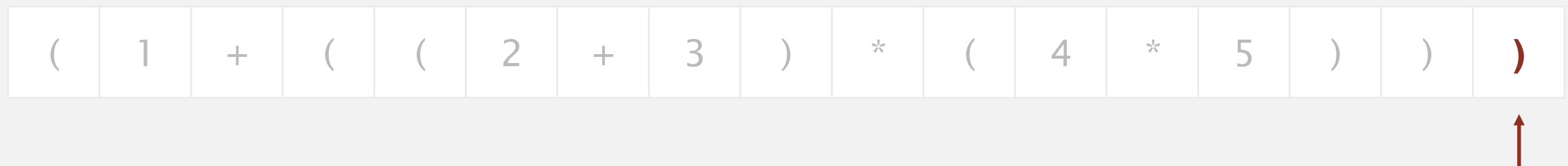
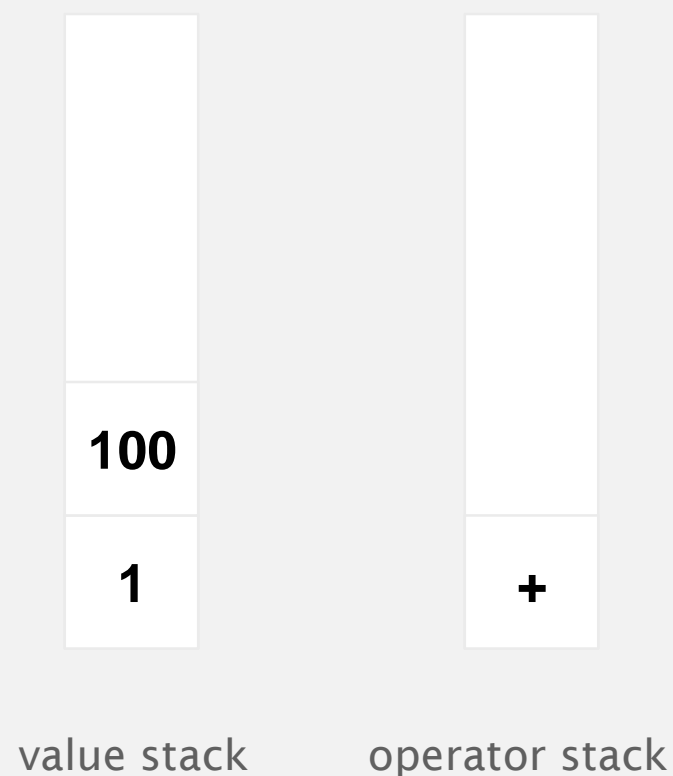
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



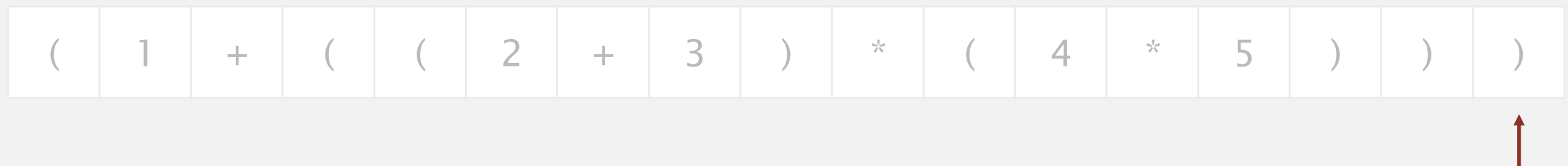
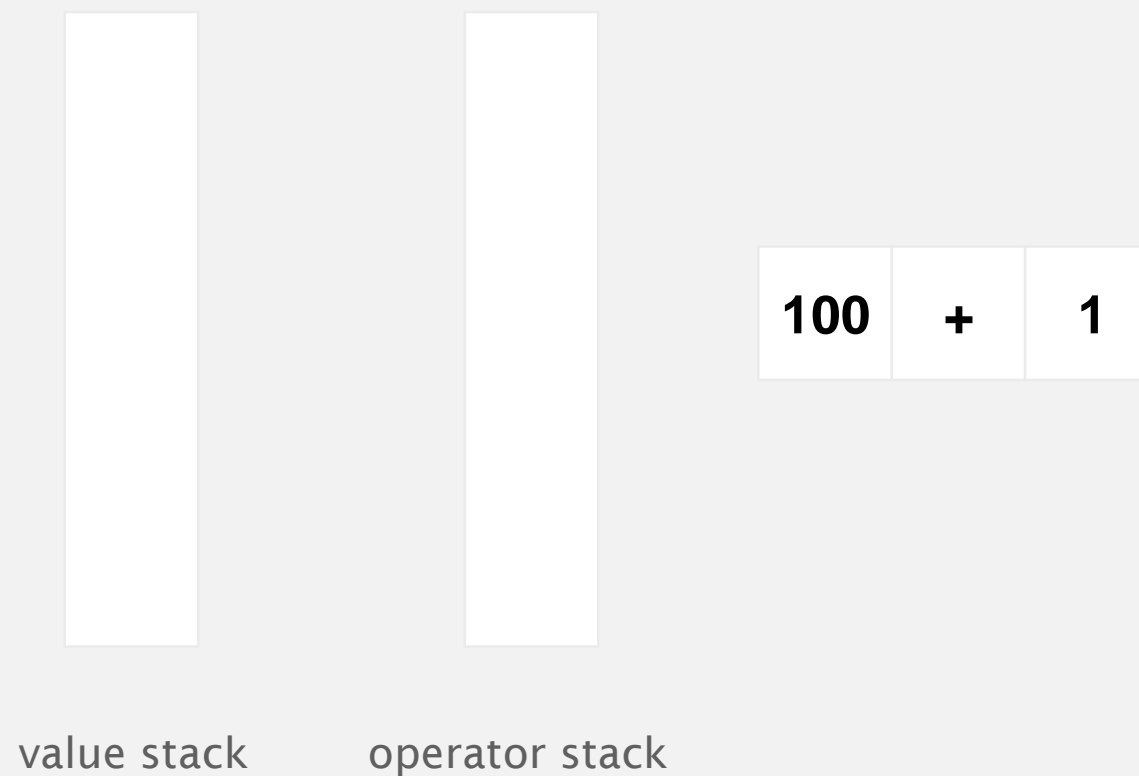
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



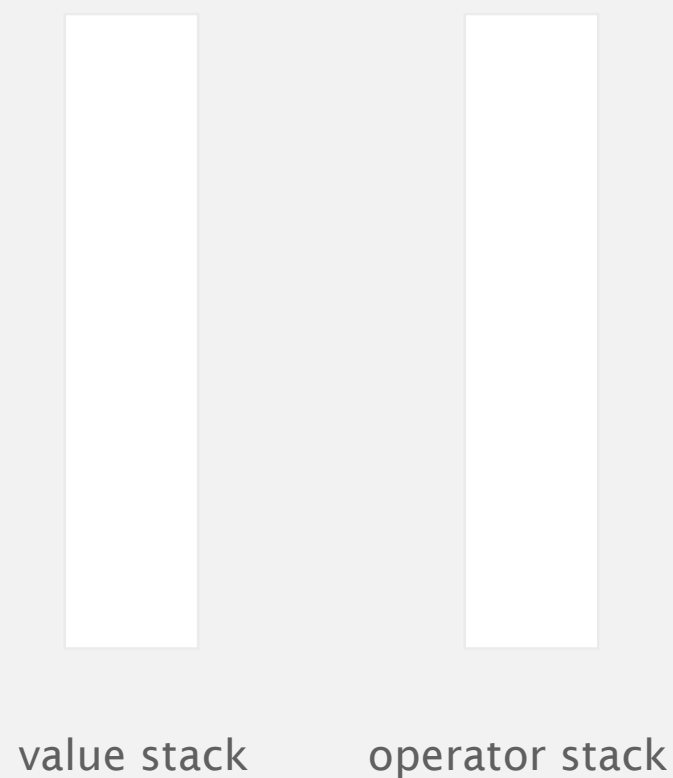
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



100	+	1	=	101
-----	---	---	---	-----

(	1	+	(	(	2	+	3	)	*	(	4	*	5	)	)	)
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



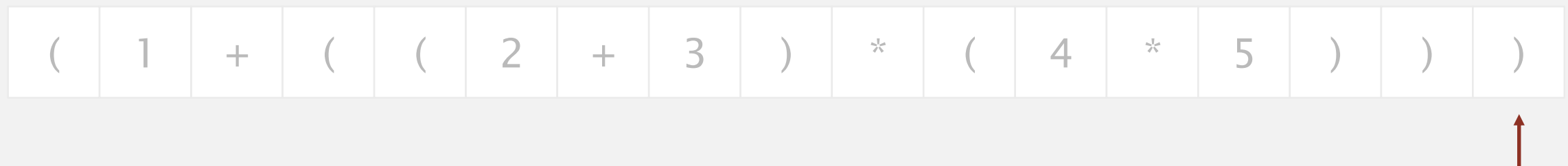
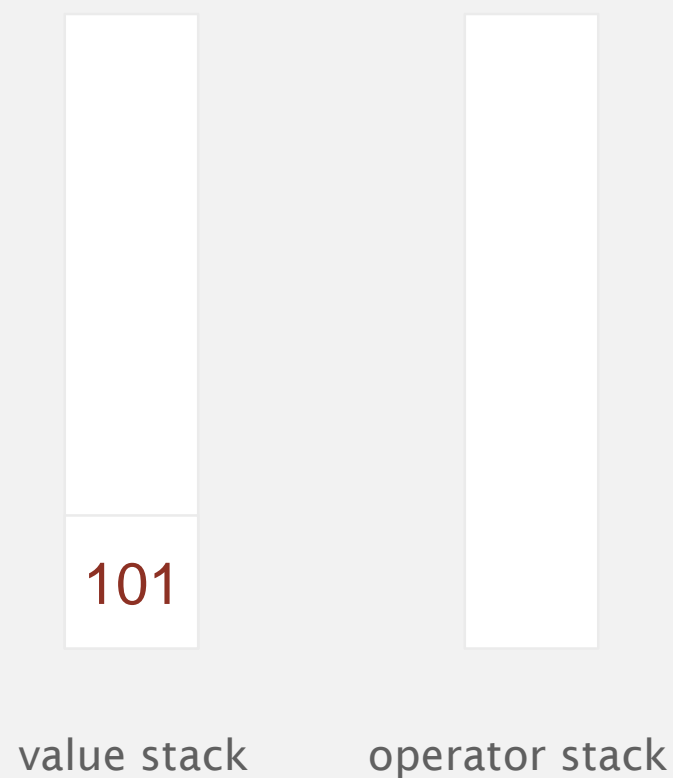
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



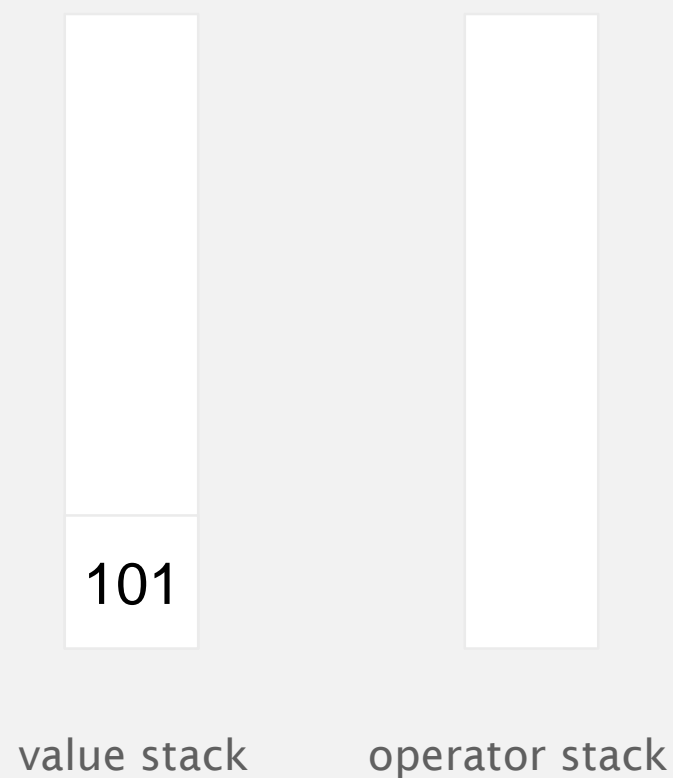
---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.



(	1	+	(	(	2	+	3	)	*	(	4	*	5	)	)	)
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---





---

**Value:** push onto the value stack.

**Operator:** push onto the operator stack.

**Left parenthesis:** ignore.

**Right parenthesis:** pop operator and two values; push the result of applying that operator to those values onto the operand stack.

101

result

(	1	+	(	(	2	+	3	)	*	(	4	*	5	)	)	)
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---