

6.5.8 Bitwise shift operators

Syntax

1 *shift-expression:*

additive-expression

shift-expression **<<** *additive-expression*

shift-expression **>>** *additive-expression*

Constraints

2 Each of the operands shall have integer type.

Semantics

3 The integer promotions are performed on each of the operands. The type of the result is that of the promoted left operand. If the value of the right operand is negative or is greater than or equal to the width of the promoted left operand, the behavior is undefined.

4 The result of **E1 << E2** is **E1** left-shifted **E2** bit positions; vacated bits are filled with zeros. If **E1** has an unsigned type, the value of the result is **E1 × 2^{E2}**, wrapped around. If **E1** has a signed type and nonnegative value, and **E1 × 2^{E2}** is representable in the result type, then that is the resulting value; otherwise, the behavior is undefined.

5 The result of **E1 >> E2** is **E1** right-shifted **E2** bit positions. If **E1** has an unsigned type or if **E1** has a signed type and a nonnegative value, the value of the result is the integral part of the quotient of **E1/2^{E2}**. If **E1** has a signed type and a negative value, the resulting value is implementation-defined.