**How to Convert a Decimal Number to Binary Using the Subtraction-Based Method:**

1. **Lay out empty slots for possible bits based on the range of your number**. Commonly used are 8 bits for numbers up to 255, but you can adjust based on the size of your number. Each dash represents a power of 2, starting from the largest on the left to the smallest on the right.

For an 8-bit number:

\_ \_ \_ \_ \_ \_ \_ \_

128 64 32 16 8 4 2 1

These represent 128, 64, 32, 16, 8, 4, 2, and 1 respectively.

1. **Starting from the leftmost bit (highest value) and working right**:
   * Compare the current decimal number to the value represented by the empty slot.
   * If the slot value is less than or equal to the decimal number:
     + Place a '1' in that slot.
     + Subtract the slot value from the decimal number.
   * If the slot value is greater than the decimal number:
     + Place a '0' in that slot.
   * Continue to the next slot using the remainder from the subtraction.
2. **Repeat this process** until you've filled all slots or the remainder becomes 0.
3. **The resulting binary number** is read from the leftmost to the rightmost filled slot.

**Convert 45 to Binary Using the Subtraction-Based Method:**

1. **Lay out empty slots for the number 45**:

\_ \_ \_ \_ \_ \_ \_ \_

128 64 32 16 8 4 2 1

Representing: 128, 64, 32, 16, 8, 4, 2, 1

1. **Begin the conversion**:
   * Is 45 ≥ 128? No. Place a 0.
   * Is 45 ≥ 64? No. Place a 0.
   * Is 45 ≥ 32? Yes. Place a 1 and subtract 32 from 45. Remaining: 13

0 0 1 \_ \_ \_ \_ \_

* + Is 13 ≥ 16? No. Place a 0.
  + Is 13 ≥ 8? Yes. Place a 1 and subtract 8 from 13. Remaining: 5

0 0 1 0 1 \_ \_ \_

* + Is 5 ≥ 4? Yes. Place a 1 and subtract 4 from 5. Remaining: 1

0 0 1 0 1 1 \_ \_

* + Is 1 ≥ 2? No. Place a 0.
  + Is 1 ≥ 1? Yes. Place a 1 and subtract 1 from 1. Remaining: 0

0 0 1 0 1 1 0 1

The binary representation for 45 using this method is: 00101101 or 101101 when omitting leading zeros.