SOURCE CODE

```c
#include <stdio.h>
#include <conio.h>

// Function to perform binary addition
int binaryAddition(int a, int b) {
    int carry = 0, result = 0, bit = 1;
    while (a != 0 || b != 0) {
        int bit_a = a % 10;
        int bit_b = b % 10;

            // Add the bits along with the carry
            int sum = bit_a + bit_b + carry;

            // Update the result
            result += (sum % 2) * bit;

            // Calculate the carry for the next bit
            carry = sum / 2;

            // Move to the next bit
            a /= 10;
            b /= 10;
            bit *= 10;
        }

        // Add the carry if exists
        result += carry * bit;
        return result;
    }

    // Function to perform logical shift left
    int logicalShiftLeft(int num) {
        return num * 10; // Equivalent to shifting left by 1 position
    }

    // Function to perform logical shift right
    int logicalShiftRight(int num) {
        return num / 10; // Equivalent to shifting right by 1 position
    }

    // Function to perform Booth's multiplication algorithm
    int boothMultiply(int multiplicand, int multiplier) {
        int accumulator = 0;
        int bitMask = 1;
```

```
45.
46.         // Iterate over each bit of the multiplier
47.         while (multiplier != 0) {
48.             // Step 2: Test Y0; if it is 1, add content of X to the
    accumulator A
49.             if (multiplier % 10 == 1) {
50.                 accumulator = binaryAddition(accumulator, multiplicand);
51.             }
52.
53.             // Step 3: Logical Shift the content of X left one position and
    content of Y right one position
54.             multiplicand = logicalShiftLeft(multiplicand);
55.             multiplier = logicalShiftRight(multiplier);
56.
57.             // Move the bit mask to the next bit
58.             bitMask *= 10;
59.         }
60.
61.         return accumulator;
62.     }
63.
64.     // Function to convert binary number to decimal
65.     int binaryToDecimal(int binary) {
66.         int decimal = 0, base = 1;
67.         while (binary != 0) {
68.             int lastDigit = binary % 10;
69.             decimal += lastDigit * base;
70.             binary /= 10;
71.             base *= 2;
72.         }
73.         return decimal;
74.     }
75.
76.     int main() {
77.         int multiplicand, multiplier;
78.         printf("Enter the multiplicand (binary): ");
79.         scanf("%d", &multiplicand);
80.         printf("Enter the multiplier (binary): ");
81.         scanf("%d", &multiplier);
82.
83.         // Step 1: Clear the accumulator (sum)
84.         int product = boothMultiply(multiplicand, multiplier);
85.
86.         printf("Product of the two binary numbers: %d (binary)\n",
    product);
87.         printf("Product in decimal: %d\n", binaryToDecimal(product));
88.
```

```
89.         getch();
90.         return 0;
91.     }
```

OUTPUT

```
Enter the multiplicand (binary): 101
Enter the multiplier (binary): 10001
Product of the two binary numbers: 1010101 (binary)
Product in decimal: 85
```

```
Enter the multiplicand (binary): 11
Enter the multiplier (binary): 10
Product of the two binary numbers: 110 (binary)
Product in decimal: 6
```

Fig: Output of Code