Program :

```
//PRN : 241106001
//Name : Amey Sangpal

#include <stdio.h>
// Function to print a number in binary (fixed width)
void printBinary(int num, int bits) {
   for (int i = bits - 1; i >= 0; i--) {
      printf("%d", (num >> i) & 1);
   }
}
// Function for arithmetic right shift on (AC, Q, Qn+1)
void arithmeticRightShift(int *AC, int *Q, int *Qn1, int n) {
   int msbAC = (*AC >> (n - 1)) & 1; // Preserve sign bit of AC
   *Qn1 = *Q & 1;                    // Save LSB of Q
   *Q = (*Q >> 1) | ((*AC & 1) << (n - 1)); // Shift Q, bring AC LSB
   *AC = (*AC >> 1) | (msbAC << (n - 1));   // Arithmetic shift AC
}
int boothsMultiplication(int M, int Q, int n) {
   int AC = 0, Qn1 = 0, count = n;
   printf("\nInitial State:\n");
   printf("AC = "); printBinary(AC, n);
   printf("  Q = "); printBinary(Q, n);
   printf("  Qn+1 = %d\n", Qn1);
   while (count > 0) {
      int Qn = Q & 1;
      // Decide operation
      if (Qn == 0 && Qn1 == 1) {
         AC = AC + M; // Add
         printf("\nOperation: AC = AC + M\n");
      } else if (Qn == 1 && Qn1 == 0) {
         AC = AC - M; // Subtract
         printf("\nOperation: AC = AC - M\n");
      } else {
         printf("\nOperation: No arithmetic (just shift)\n");
```

```
        }
        // Print before shift
        printf("Before Shift -> AC = "); printBinary(AC, n);
        printf("  Q = "); printBinary(Q, n);
        printf("  Qn+1 = %d\n", Qn1);
        // Arithmetic Right Shift
        arithmeticRightShift(&AC, &Q, &Qn1, n);

        // Print after shift
        printf("After Shift  -> AC = "); printBinary(AC, n);
        printf("  Q = "); printBinary(Q, n);
        printf("  Qn+1 = %d\n", Qn1);
        count--;
    }
    // Combine AC and Q for final result
    int result = (AC << n) | (Q & ((1 << n) - 1));

    printf("\nFinal Product (Binary): ");
    printBinary(result, 2 * n);
    printf("\nFinal Product (Decimal): %d\n", result);
    return result;
}
int main() {
    int M, Q, n;
    printf("Enter multiplicand: ");
    scanf("%d", &M);
    printf("Enter multiplier: ");
    scanf("%d", &Q);
    printf("Enter number of bits: ");
    scanf("%d", &n);
    boothsMultiplication(M, Q, n);
    return 0;
}
```

Output

```
Enter multiplicand: 6
Enter multiplier: 4
Enter number of bits: 4

Initial State:
AC = 0000  Q = 0100  Qn+1 = 0

Operation: No arithmetic (just shift)
Before Shift -> AC = 0000  Q = 0100  Qn+1 = 0
After Shift  -> AC = 0000  Q = 0010  Qn+1 = 0

Operation: No arithmetic (just shift)
Before Shift -> AC = 0000  Q = 0010  Qn+1 = 0
After Shift  -> AC = 0000  Q = 0001  Qn+1 = 0

Operation: AC = AC - M
Before Shift -> AC = 1010  Q = 0001  Qn+1 = 0
After Shift  -> AC = 1101  Q = 0000  Qn+1 = 1

Operation: AC = AC + M
Before Shift -> AC = 0011  Q = 0000  Qn+1 = 1
After Shift  -> AC = 0001  Q = 1000  Qn+1 = 0

Final Product (Binary): 00011000
Final Product (Decimal): 24


=== Code Execution Successful ===
```