**Booth’s algorithm of multiplication of two numbers:**

**Algorithm:**

Step 1: Start.

Step 2: Initialize and take number inputs in decimal.

Step 3: Initialize array of specific bit size Q, M and Mcomp. Also initialize accumulator to 0, an integer a = 0 and counter = no. of bits.

Step 4: Convert the first decimal number to binary. If the number is negative, compute the two’s complement for the number and store it in Q.

Step 5: Convert the first decimal number to binary. If the number is negative, compute the two’s complement for the number and store it in Q.

Step 6: Find the 2’s complement of M and store it in Mcomp.

Step 7: For ‘n’ bits of data, if Q[n-1].a = 01, perform A🡨A+M and right shift accumulator, Q and ‘a’. else if Q[n-1].a = 10, perform A🡨A-M and right shift accumulator, Q and ‘a’. else right shift accumulator, Q and ‘a’.

Step 8: count --.

Step 9: if count > 0, go to step 7.

Step 10: Stop.

**Source code:**

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <math.h>*

*#define SIZE 5*

*void decimalToBinary(int n, int \*binary, int size) {*

*if (n < 0) {*

*n = abs(n);*

*}*

*for (int i = size - 1; i >= 0; i--) {*

*binary[i] = n % 2;*

*n = n / 2;*

*}*

*}*

*void twosComplement(int \*binary, int size) {*

*int carry = 1;*

*for (int i = 0; i < size; i++) {*

*binary[i] = binary[i] == 0 ? 1 : 0;*

*}*

*for (int i = size - 1; i >= 0; i--) {*

*binary[i] = binary[i] + carry;*

*if (binary[i] == 2) {*

*binary[i] = 0;*

*carry = 1;*

*} else {*

*carry = 0;*

*}*

*}*

*}*

*void arithmeticRightShift(int \*binary1, int \*binary2, int \*a, int size) {*

*\*a = binary2[size - 1];*

*for (int i = size - 1; i > 0; i--) {*

*binary2[i] = binary2[i - 1];*

*}*

*binary2[0] = binary1[size - 1];*

*int msb = binary1[0];*

*for (int i = size - 1; i > 0; i--) {*

*binary1[i] = binary1[i - 1];*

*}*

*binary1[0] = msb;*

*}*

*void addTwoBinaries(int \*binary1, int \*binary2, int size) {*

*int carry = 0;*

*for (int i = size - 1; i >= 0; i--) {*

*binary1[i] = binary1[i] + binary2[i] + carry;*

*if (binary1[i] == 2) {*

*binary1[i] = 0;*

*carry = 1;*

*} else if (binary1[i] == 3) {*

*binary1[i] = 1;*

*carry = 1;*

*} else {*

*carry = 0;*

*}*

*}*

*}*

*void printBinary(int \*binary, int size) {*

*for (int i = 0; i < size; i++) {*

*printf("%d", binary[i]);*

*}*

*}*

*void printRow(int count, int \*accumulator, int \*temp, int a, const char\* operation) {*

*printBinary(accumulator, SIZE);*

*printf(" | ");*

*printBinary(temp, SIZE);*

*printf(" | %d | %d | %s\n", a, count, operation);*

*}*

*int main() {*

*int x, y, a = 0, count = SIZE;*

*int first[SIZE], second[SIZE], accumulator[SIZE] = {0}, complementSecond[SIZE], temp[SIZE];*

*printf("Enter the first number: ");*

*scanf("%d", &x);*

*printf("Enter the second number: ");*

*scanf("%d", &y);*

*decimalToBinary(x, first, SIZE);*

*if (x < 0) {*

*twosComplement(first, SIZE);*

*}*

*printf("First number in binary: ");*

*printBinary(first, SIZE);*

*printf("\n");*

*decimalToBinary(y, second, SIZE);*

*if (y < 0) {*

*twosComplement(second, SIZE);*

*}*

*printf("Second number in binary: ");*

*printBinary(second, SIZE);*

*printf("\n\n");*

*for (int i = 0; i < SIZE; i++) {*

*complementSecond[i] = second[i];*

*}*

*twosComplement(complementSecond, SIZE);*

*for (int i = 0; i < SIZE; i++) {*

*temp[i] = first[i];*

*}*

*printf("| A | Q | Q-1 | COUNT | Remarks\n");*

*printf("|-------|-------|-----| ----- |-----------\n");*

*printRow(count, accumulator, temp, a, "Initialization");*

*while (count > 0) {*

*if ((temp[SIZE - 1] == 0) && (a == 1)) {*

*addTwoBinaries(accumulator, second, SIZE);*

*printRow(count, accumulator, temp, a, "Addition");*

*arithmeticRightShift(accumulator, temp, &a, SIZE);*

*count--;*

*printRow(count, accumulator, temp, a, "Shift");*

*printf("\n");*

*} else if ((temp[SIZE - 1] == 1) && (a == 0)) {*

*addTwoBinaries(accumulator, complementSecond, SIZE);*

*printRow(count, accumulator, temp, a, "Subtraction");*

*arithmeticRightShift(accumulator, temp, &a, SIZE);*

*count--;*

*printRow(count, accumulator, temp, a, "Shift");*

*printf("\n");*

*} else {*

*arithmeticRightShift(accumulator, temp, &a, SIZE);*

*count--;*

*printRow(count, accumulator, temp, a, "Shift");*

*printf("\n");*

*}*

*}*

*printf("Result after Booth's multiplication:\n");*

*printBinary(accumulator, SIZE);*

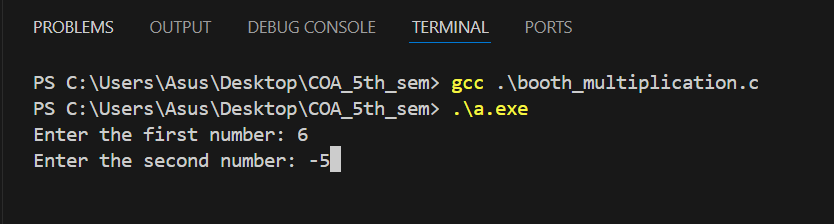
*printBinary(temp, SIZE);*

*return 0;*

*}*

**Sample input/output:**

* Input of 6 and -5

****

* Sample output for above input

**A screenshot of a computer program

Description automatically generated**