#include <stdio.h>

#include <stdlib.h>

void binary(int a, int b, int \*anum, int \*anumcp, int \*bnum, int \*acomp, int \*bcomp) {

int com[5] = {1, 0, 0, 0, 0};

int r, r2;

int i;

a = abs(a);

b = abs(b);

for (i = 0; i < 5; i++) {

r = a % 2;

a = a / 2;

r2 = b % 2;

b = b / 2;

anum[i] = r;

anumcp[i] = r;

bnum[i] = r2;

if (r2 == 0) {

bcomp[i] = 1;

}

if (r == 0) {

acomp[i] = 1;

}

}

// Two's complement

int c = 0;

int res[5];

for (i = 0; i < 5; i++) {

res[i] = com[i] + bcomp[i] + c;

if (res[i] >= 2) {

c = 1;

} else {

c = 0;

}

res[i] = res[i] % 2;

}

for (i = 4; i >= 0; i--) {

bcomp[i] = res[i];

}

}

void add(int \*rem, int \*anumcp, int \*num) {

int res[5];

int c = 0;

int i;

for (i = 0; i < 5; i++) {

res[i] = rem[i] + num[i] + c;

if (res[i] >= 2) {

c = 1;

} else {

c = 0;

}

res[i] = res[i] % 2;

}

for (i = 4; i >= 0; i--) {

rem[i] = res[i];

printf("%d", rem[i]);

}

printf(":");

for (i = 4; i >= 0; i--) {

printf("%d", anumcp[i]);

}

}

void shl(int \*rem, int \*anumcp) {

int temp = anumcp[4];

int i;

for (i = 4; i > 0; i--) {

rem[i] = rem[i - 1];

anumcp[i] = anumcp[i - 1];

}

rem[0] = temp;

anumcp[0] = 0;

printf("\nSHIFT LEFT:");

for (i = 4; i >= 0; i--) {

printf("%d", rem[i]);

}

printf(":");

for (i = 4; i >= 0; i--) {

printf("%d", anumcp[i]);

}

}

int main() {

int i, a, b, s = 0;

int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0};

int acomp[5] = {0}, bcomp[5] = {0}, rem[5] = {0};

printf("\t\tRESTORING DIVISION ALGORITHM\n");

printf("Enter two numbers to divide (both must be less than 16):\n");

do {

printf("Enter A: ");

scanf("%d", &a);

printf("Enter B: ");

scanf("%d", &b);

} while (a >= 16 || b >= 16);

printf("\nExpected Quotient = %d", a / b);

printf("\nExpected Remainder = %d\n", a % b);

if (a \* b < 0) {

s = 1;

}

binary(a, b, anum, anumcp, bnum, acomp, bcomp);

printf("\n\nUnsigned Binary Equivalents are:\n");

printf("A = ");

for (i = 4; i >= 0; i--) {

printf("%d", anum[i]);

}

printf("\nB = ");

for (i = 4; i >= 0; i--) {

printf("%d", bnum[i]);

}

printf("\nB' + 1 = ");

for (i = 4; i >= 0; i--) {

printf("%d", bcomp[i]);

}

printf("\n\n-->");

shl(rem, anumcp);

for (i = 0; i < 5; i++) {

printf("\n-->");

printf("\nSUB B: ");

add(rem, anumcp, bcomp);

if (rem[4] == 1) {

printf("\n-->RESTORE");

printf("\nADD B: ");

anumcp[0] = 0;

add(rem, anumcp, bnum);

} else {

anumcp[0] = 1;

}

if (i < 4) {

shl(rem, anumcp);

}

}

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

printf("\nSign of the result = %d", s);

printf("\nRemainder is = ");

for (i = 4; i >= 0; i--) {

printf("%d", rem[i]);

}

printf("\nQuotient is = ");

for (i = 4; i >= 0; i--) {

printf("%d", anumcp[i]);

}

return 0;

}