//Program On Booth Algorithm

#include<stdio.h>

#include<stdlib.h>

#include<malloc.h>

#include<ctype.h>

#include<string.h>

void \_2comp(int negm[],int a[],int n)

{

int c=1;

int i;

for(i=0;i<n;i++)

{

if(a[i]==0)

{

negm[i]=1;

}

else

{

negm[i]=0;

}

}

for(i=0;i<n;i++)

{

if(c==1)

{

if(negm[i]==1)

{

negm[i]=0;

c=1;

}

else

{

negm[i]=1;

c=0;

}

}

}

}

/////

void sum(int a[],int b[],int n)

{

int i,c=0;

for(i=0;i<n;i++)

{

if(c==1)

{

if(a[i]==1 && b[i]==1)

{

a[i]=1;

c=1;

}

else if(a[i]==0 && b[i]==0)

{

a[i]=1;

c=0;

}

else if((a[i]==1 && b[i]==0) || (a[i]==0 && b[i]==1))

{

a[i]=0;

c=1;

}

}

else

{

if(a[i]==1 && b[i]==1)

{

a[i]=0;

c=1;

}

else if(a[i]==0 && b[i]==0)

{

a[i]=0;

c=0;

}

else if((a[i]==1 && b[i]==0) || (a[i]==0 && b[i]==1))

{

a[i]=1;

}

}

}

}

/////

void shift(int a[],int q[],int n,int \*q0)

{

int i=0;

\*q0=q[0];

q[0]=q[1];

while(i<=n-2)

{

q[i]=q[i+1];

i++;

}

q[n-1]=a[0];

i=0;

while(i<=n-2)

{

a[i]=a[i+1];

i++;

}

}

/////

int result(int a[],int q[],int n)

{

int ans=0;

int i=0,j,r;

int h;

while(i<n)

{

h=1;

for(j=1;j<=i;j++)

{

h=h\*2;

}

r=j;

ans+=q[i]\*h;

i++;

}

i=0;

while(i<n)

{

h=1;

for(j=1;j<=r;j++)

{

h=h\*2;

}

r++;

ans+=a[i]\*h;

i++;

}

return ans;

}

/////

void comp(int a[],int q[],int n)

{

int i,c=1;

for(i=0;i<n;i++)

{

if(a[i]==0)

{

a[i]=1;

}

else

{

a[i]=0;

}

}

for(i=0;i<n;i++)

{

if(q[i]==0)

{

q[i]=1;

}

else

{

q[i]=0;

}

}

for(i=0;i<n;i++)

{

if(q[i]==1 && c==1)

{

q[i]=0;

c=1;

}

else if(q[i]==0 && c==1)

{

q[i]=1;

c=0;

}

}

for(i=0;i<n;i++)

{

if(a[i]==1 && c==1)

{

a[i]=0;

c=1;

}

else if(a[i]==0 && c==1)

{

a[i]=1;

c=0;

}

}

}

/////

void two\_comp\_single(int p[],int n)

{

int c=1;

int i;

for(i=0;i<n;i++)

{

if(p[i]==0)

{

p[i]=1;

}

else

{

p[i]=0;

}

}

for(i=0;i<n;i++)

{

if(c==1)

{

if(p[i]==1)

{

p[i]=0;

c=1;

}

else

{

p[i]=1;

c=0;

}

}

}

}

/////

void dec2bin(int x,int y,int m[],int q[],int \*n,int a[])

{

int i,k=0,f1=0,f2=0,ref;

ref=\*n;

if(x<0)

{

x=abs(x);

f1=1;

}

if(y<0)

{

y=abs(y);

f2=1;

}

while(x!=0 || y!=0)

{

if(k>ref-1)

{

\*n=\*n+1;

}

m[k]=x%2;

q[k]=y%2;

x=x/2;

y=y/2;

k++;

}

\*n=\*n+1;

q[\*n-1]=0;

m[\*n-1]=0;

for(i=0;i<\*n;i++)

{

a[i]=0;

}

if(f1==1)

{

two\_comp\_single(m,\*n);

}

if(f2==1)

{

two\_comp\_single(q,\*n);

}

}

/////

int main()

{

int x,y,q0=0;

int ans;

int i,n=16; //Use size of 'n' for larger calc higher the value of 'n' higher the cal //

int m[n],q[n],a[n];

printf("\nBOOTH ALGORITHM : ");

for(i=0;i<n;i++)

{

q[i]=0;

m[i]=0;

}

printf("\nEnter the two numbers to multiply : ");

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

dec2bin(x,y,m,q,&n,a);

printf("\nBinary Form Of 'x' ie %d : M -> ",x);

for(i=n-1;i>=0;i--)

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

printf("\nBinary Form Of 'y' ie %d : Q -> ",y);

for(i=n-1;i>=0;i--)

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

printf("\n");

int negm[n];

int j=n,k=1;

\_2comp(negm,m,n);

printf("\nNeg M ie (-M) -> ");

for(i=n-1;i>=0;i--)

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

while(j--)

{

if(q[0]==1 && q0==0)

{

sum(a,negm,n);

}

else if(q[0]==0 && q0==1)

{

sum(a,m,n);

}

shift(a,q,n,&q0);

}

if(a[n-1]==1)

{

comp(a,q,n);

k=-1;

}

ans=result(a,q,n);

ans\*=k;

printf("\n\nMultiplication Of %d & %d is : %d",x,y,ans);

return 0;

}