Program:

#include<stdio.h>

#include<math.h>

void binary();

void sum(int num[]);

void arithmetic\_shift();

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

int first\_number[5] = {0}, second\_number[5] = {0}, anumcp[5] = {0};

int compare\_num1[5] = {0}, compare\_num2[5] = {0}, product[5] = {0}, result[5] = {0};

int num1 = 0, num2 = 0, num3 = 0;

int m = 0, n = 0;

int main()

{

int count, x = 0;

printf("Enter Two Numbers to Multiply (Less Than 16)\n");

do

{

printf("Enter A:\t");

scanf("%d", &num1);

printf("Enter B:\t");

scanf("%d", &num2);

}while(num1 >=16 || num2 >=16);

printf("\nExpected Product of %d \* %d = %d", num1, num2, num1 \* num2);

binary();

printf("\n\nBinary Equivalents\n");

printf("\nA:\t");

for(count = 4; count >= 0; count--)

{

printf("%d", first\_number[count]);

}

printf("\nB:\t");

for(count = 4; count >= 0; count--)

{

printf("%d", second\_number[count]);

}

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

for(count = 4; count >= 0; count--)

{

printf("%d", compare\_num2[count]);

}

printf("\n");

for(count = 0; count < 5; count++)

{

if(first\_number[count] == x)

{

printf("\n-->");

arithmetic\_shift();

x = first\_number[count];

}

else if(first\_number[count] == 1 && x == 0)

{

printf("\n-->");

printf("\nSUB B: ");

sum(compare\_num2);

arithmetic\_shift();

x = first\_number[count];

}

else

{

printf("\n-->");

printf("\nADD B: ");

sum(second\_number);

arithmetic\_shift();

x = first\_number[count];

}

}

printf("\nProduct:\t");

for(count = 4; count >= 0; count--)

{

printf("%d", product[count]);

}

for(count = 4; count >= 0; count--)

{

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

}

printf("\n");

return 0;

}

void binary()

{

m = fabs(num1);

n = fabs(num2);

int r2, remainder, count, temp;

for(count = 0; count < 5; count++)

{

remainder = m % 2;

m = m / 2;

r2 = n % 2;

n = n / 2;

first\_number[count] = remainder;

anumcp[count] = remainder;

second\_number[count] = r2;

if(r2 == 0)

{

compare\_num2[count] = 1;

}

if(remainder == 0)

{

compare\_num1[count] =1;

}

}

num3 = 0;

for(count = 0; count < 5; count++)

{

result[count] = comparison[count]+ compare\_num2[count] + num3;

if(result[count] >= 2)

{

num3 = 1;

}

else

{

num3 = 0;

}

result[count] = result[count] % 2;

}

for(count = 4; count >= 0; count--)

{

compare\_num2[count] = result[count];

}

if(num1 < 0)

{

num3 = 0;

for(count = 4; count >= 0; count--)

{

result[count] = 0;

}

for(count = 0; count < 5; count++)

{

result[count] = comparison[count] + compare\_num1[count] + num3;

if(result[count] >= 2)

{

num3 = 1;

}

else

{

num3 = 0;

}

result[count] = result[count] % 2;

}

for(count = 4; count >= 0; count--)

{

first\_number[count] = result[count];

anumcp[count] = result[count];

}

}

if(num2 < 0)

{

for(count = 0; count < 5; count++)

{

temp = second\_number[count];

second\_number[count] = compare\_num2[count];

compare\_num2[count] = temp;

}

}

}

void sum(int num[])

{

int count;

num3 = 0;

for(count = 0; count < 5; count++)

{

result[count] = product[count] + num[count] + num3;

if(result[count] >= 2)

{

num3 = 1;

}

else

{

num3 = 0;

}

result[count] = result[count] % 2;

}

for(count = 4; count >= 0; count--)

{

product[count] = result[count];

printf("%d", product[count]);

}

printf(":");

for(count = 4; count >= 0; count--)

{

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

}

}

void arithmetic\_shift()

{

int x = product[4], y = product[0], count;

for(count = 1; count < 5 ; count++)

{

product[count - 1] = product[count];

}

product[4] = x;

for(count = 1; count < 5; count++)

{

anumcp[count - 1] = anumcp[count];

}

anumcp[4] = y;

printf("\nArithmetic Shift");

for(count = 4; count >= 0; count--)

{

printf("%d", product[count]);

}

printf(":");

for(count = 4; count >= 0; count--)

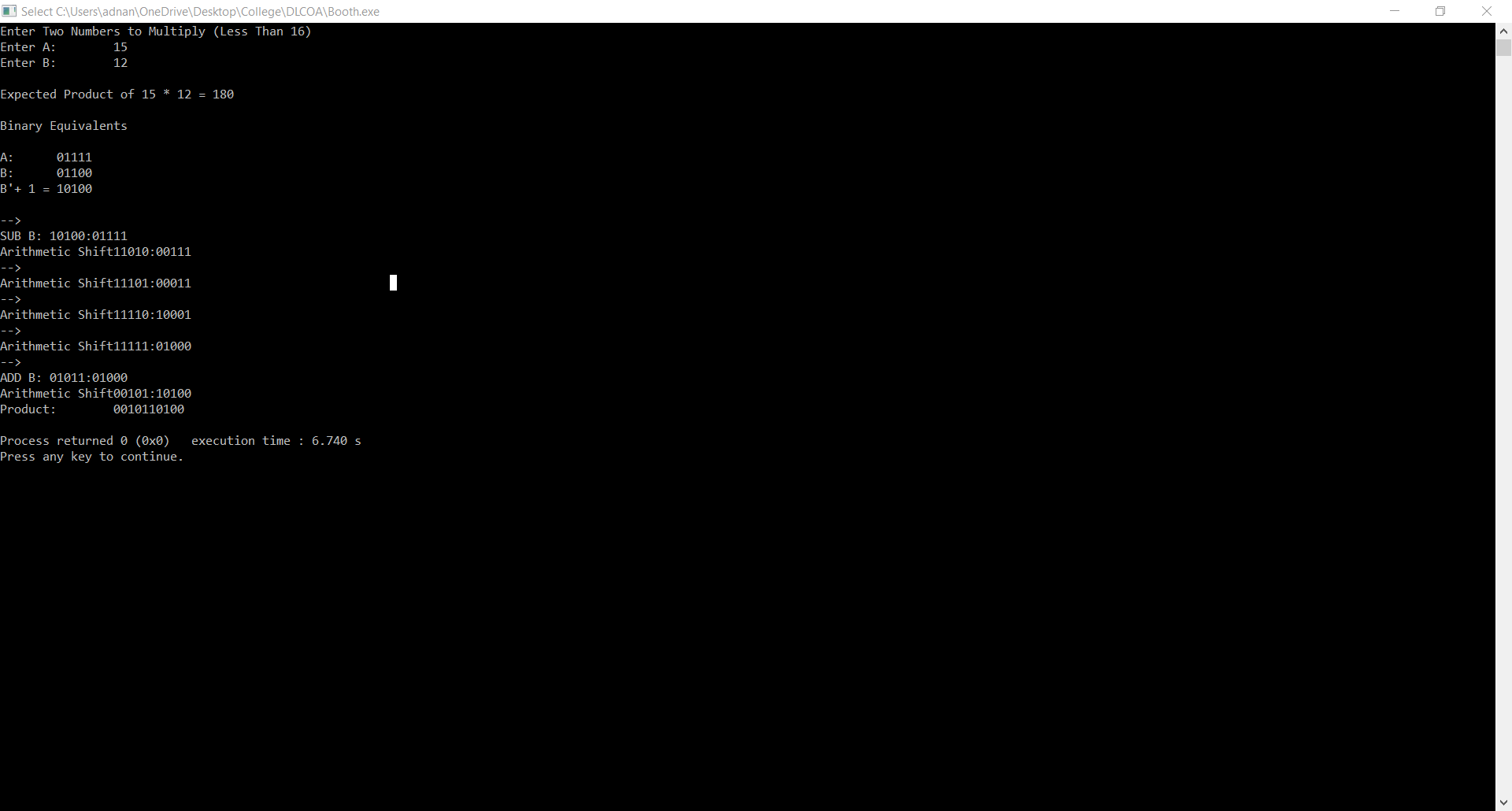
{

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

}

}

Output:



Program:

#include<stdio.h>

#include<math.h>

int getsize(int x)

{

int c;

if(x<=1)

c = 2;

else if(x < 4)

c = 2;

else if(x< 8)

c = 3;

else if(x< 16)

c = 4;

else if(x< 32)

c = 5;

else if(x< 64)

c = 6;

else if(x< 128)

c = 7;

else if(x< 256)

c = 8;

else if(x< 512)

c = 9;

return c;

}

int max(int x,int y)

{

if(x< y)

return(y);

else

return(x);

}

void main()

{

int B,Q,Z,M,c,c1,e,f,g,h,i,j,x,y,ch,in,S,G,P;

int a[24],b[12],b1[12],q[12],carry=0,count=0,option;

long num;

do

{

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

printf("¦\t\tPROGRAM FOR DIVISION\t\t¦\n");

printf("¦-----------------------------------------------¦");

printf("\n\nENTER DIVIDEND\t: ");

scanf("%d",&Q);

y = getsize(Q);

printf("ENTER DIVISOR\t: ");

scanf("%d",&M);

x = getsize(M);

Z = max(x,y);

printf("\n\tTOTAL BITS CONSIDERED FOR RESULT => %d",2\*Z+1);

printf("\n\tINITiALLY A IS RESET TO ZERO:");

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

printf("%d ",a[i]=0);

for(i=Z;i>=0;i--)

{

b1[i] = b[i] = M%2;

M = M/2;

b1[i] = 1-b1[i];

}

carry = 1;

for(i=Z;i>=0;i--)

{

c1 = b1[i]^carry;

carry = b1[i]&&carry;

b1[i]=c1;

}

for(i=2\*Z;i>Z;i--)

{

a[i] = Q%2;

Q = Q/2;

}

printf("\n\n\tDivisor\t\t(M)\t: ");

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

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

printf("\n\t2'C Divisor\t(M)\t: ");

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

printf("%d ",b1[i]);

printf("\n\tDividend\t(Q)\t: ");

for(i=Z+1;i<=2\*Z;i++)

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

printf("\n\n\tBITS CONSIDERED:[ A ] [ M ]");

printf("\n\t\t\t");

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

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

printf(" ");

for(i=Z+1;i<=2\*Z;i++)

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

count = Z;

do{

for(i=0;i< 2\*Z;i++)

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

printf("\n\nLeft Shift\t\t");

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

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

printf(" ");

for(i=Z+1;i< 2\*Z;i++)

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

carry=0;

for(i=Z;i>=0;i--)

{

S=a[i]^(b1[i]^carry);

G=a[i]&&b1[i];

P=a[i]^b1[i];

carry=G||(P&&carry);

a[i]=S ;

}

printf("\nA< -A-M \t\t");

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

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

printf(" ");

for(i=Z+1;i< 2\*Z;i++)

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

ch=a[0];

printf("\nBIT Q:%d",ch);

switch (ch)

{

case 0: a[2\*Z]=1;

printf(" Q0< -1\t\t");

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

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

printf(" ");

for(i=Z+1;i<=2\*Z;i++)

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

break;

case 1: a[2\*Z]=0;

printf(" Q0< -0\t\t");

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

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

printf(" ");

for(i=Z+1;i< 2\*Z;i++)

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

carry=0;

for(i=Z;i>=0;i--)

{

S=a[i]^(b[i]^carry);

G=a[i]&&b[i];

P=a[i]^b[i];

carry=G||(P&&carry);

a[i]=S ;

}

printf("\nA< -A+M");

printf("\t\t\t");

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

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

printf(" ");

for(i=Z+1;i<=2\*Z;i++)

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

break;

}

count--;

}while(count!=0);

num=0;

printf("\n\t\t< < QUOTIENT IN BITS>> :");

for(i=Z+1;i<=2\*Z;i++)

{

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

num=num+pow(2,2\*Z-i)\*a[i];

}

printf("\n\t\tOUOTIENT IN DECIMAL :%d",num);

num=0;

printf("\n\t\t< < REMAINDER IN BITS>>:");

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

{

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

num=num+pow(2,Z-i)\*a[i];

}

printf("\n\t\tREMAINDER IN DECIMAL :%d",num);

getche();

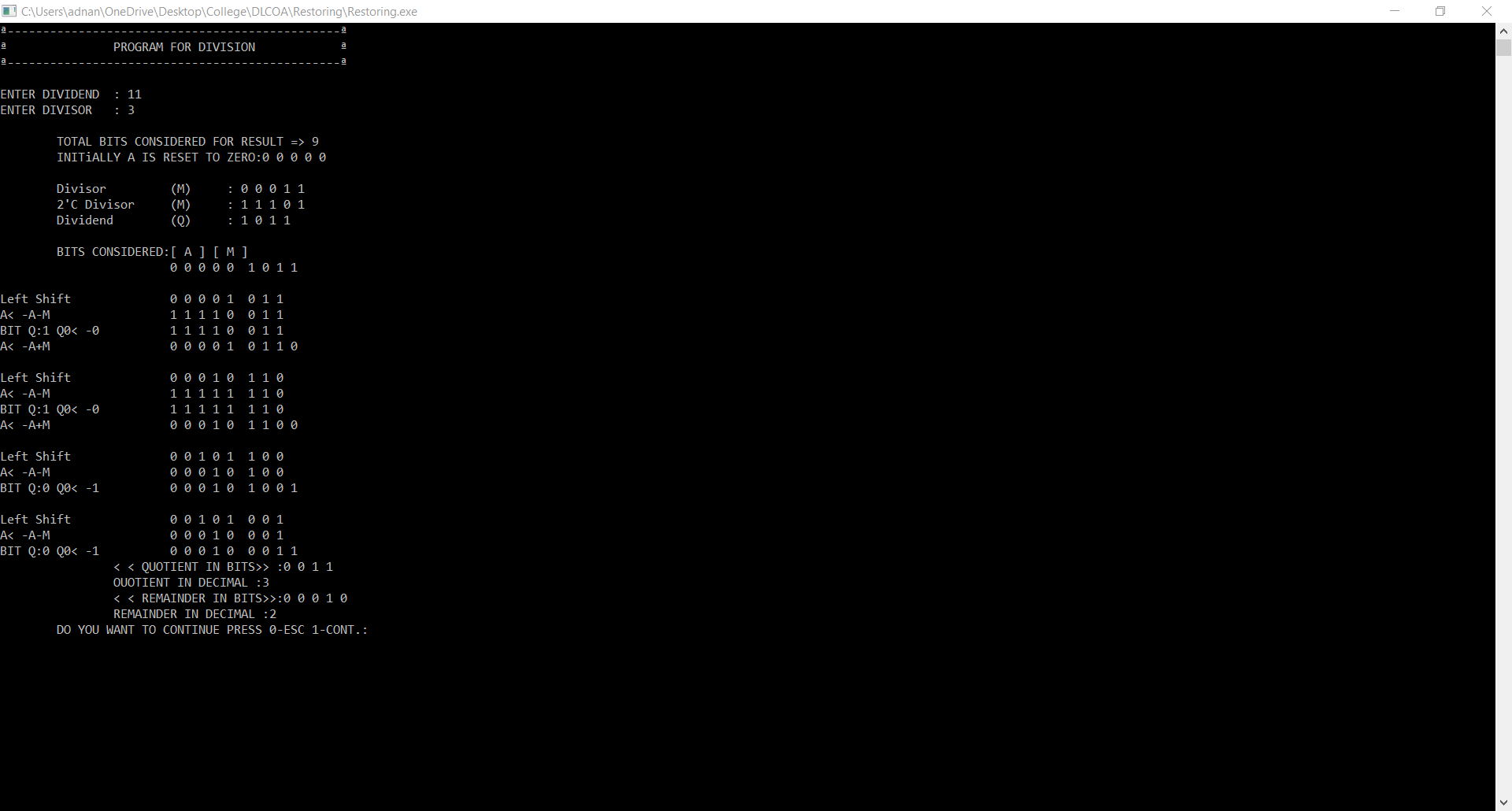
printf("\n\tDO YOU WANT TO CONTINUE PRESS 0-ESC 1-CONT.:");

scanf("%d",&option);

}while(option!=0);

}

Output:



Program:

#include<stdio.h>

void main()

{

int a[10],m[10],q[10],i,j,c=0,b[10],mc[10],z=0,s[10],x[10];

printf("\nEnter Divisor [M] : ");

for(i=4;i>=1;i--)

{

scanf("%d",&m[i]);

mc[i]=!m[i];

a[i]=0, x[i]=0;

}

x[1]=1, x[5]=0, m[5]=0, mc[5]=1;

for(i=1;i<=5;i++)

{

s[i]=x[i]^mc[i]^z;

c=(x[i]&&mc[i])||(x[i]&&z)||(mc[i]&&z);

z=c;

mc[i]=s[i];

}

printf("\n\nEnter Divident [Q] : ");

for(i=4;i>=1;i--)

{

scanf("%d",&q[i]);

}

printf("\n\n\n Step \t\t Action Performed \t\t A \t\tQ\n");

printf("\n\n 0\t\t Initialization\t 0 0 0 0 0 \t ");

for(i=4;i>=1;i--)

{

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

}

for(j=1;j<=4;j++)

{

printf("\n\n\n %d",j);

printf("\t\t Left Shift \t\t ");

for(i=5;i>=2;i--)

{

a[i]=a[i-1];

}

a[1]=q[4];

for(i=4;i>=2;i--)

{

q[i]=q[i-1];

}

for(i=5;i>=1;i--)

{

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

}

printf("\t ");

for(i=4;i>=2;i--)

{

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

}

if(a[5]==0)

{

z=0;

for(i=1;i<=5;i++)

{

s[i]=a[i]^mc[i]^z;

c=(a[i]&&mc[i])||(a[i]&&z)||(mc[i]&&z);

z=c;

a[i]=s[i];

}

if(a[5]==1)

{

q[1]=0;

}

else

{

q[1]=1;

}

printf("\n\n\t\t a = a-m\t\t ");

for(i=5;i>=1;i--)

{

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

}

printf("\t ");

for(i=4;i>=1;i--)

{

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

}

}

else

{

z=0;

for(i=1;i<=5;i++)

{

s[i]=a[i]^m[i]^z;

c=(a[i]&&m[i])||(a[i]&&z)||(m[i]&&z);

z=c;

a[i]=s[i];

}

if(a[5]==1)

{

q[1]=0;

}

else

{

q[1]=1;

}

printf("\n\n\t\t a = a+m\t\t ");

for(i=5;i>=1;i--)

{

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

}

printf("\t ");

for(i=4;i>=1;i--)

{

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

}

}

}

if(a[5]==1)

{

printf("\n\n\n 5");

for(i=1;i<=5;i++)

{

s[i]=a[i]^m[i]^z;

c=(a[i]&&m[i])||(a[i]&&z)||(m[i]&&z);

z=c;

a[i]=s[i];

}

printf("\t\t a = a+m\t\t ");

for(i=5;i>=1;i--)

{

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

}

printf("\t ");

for(i=4;i>=1;i--)

{

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

}

}

printf("\n\n\n\n\nQuotient [Q] :");

for(i=4;i>=1;i--)

{

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

}

printf("\n\n\n\nRemainder [A] :");

for(i=4;i>=1;i--)

{

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

}

}

Output:

: 