18.Write a program to perform Restoring Division of two numbers using any high level language.

AIM:

To perform restoring division of two numbers using c program

Theory:

Restoring division:

1. Restoring division operates on fixed-point fractional numbers and depends on the following assumptions: The following division methods are all based on the form *Q* = A/ *M* where

- *Q* = Quotient
- *A* = Numerator (dividend)
- *M* = Denominator (divisor).

Algorithm:

1. Start
2. Quotient = 0, Remainder =0 and Sign = 0
3. Ask the user to enter two decimal numbers: n1, n2
4. Convert their absolute values into binary and store them in arrays num1 and num2
5. Two's complement num2 and store as ncom
6. Create a copy of num1 as ncopy
7. If the product is negative, set sign = 1
8. Shift left Remainder : ncopy; counter = 0
9. Add ncom to Remainder
    1. Set LSB of ncopy as 0.
    2. If result is negative, restore the remainder
    3. Otherwise, Set LSB of ncopy as 1.
10. If counter < bits in num1, Shift left Remainder : ncopy
11. counter = counter + 1
12. Repeat 9, 10 and 11 until all bits of num1 is traced
13. Display final result as Sign, Remainder: ncopy
14. End

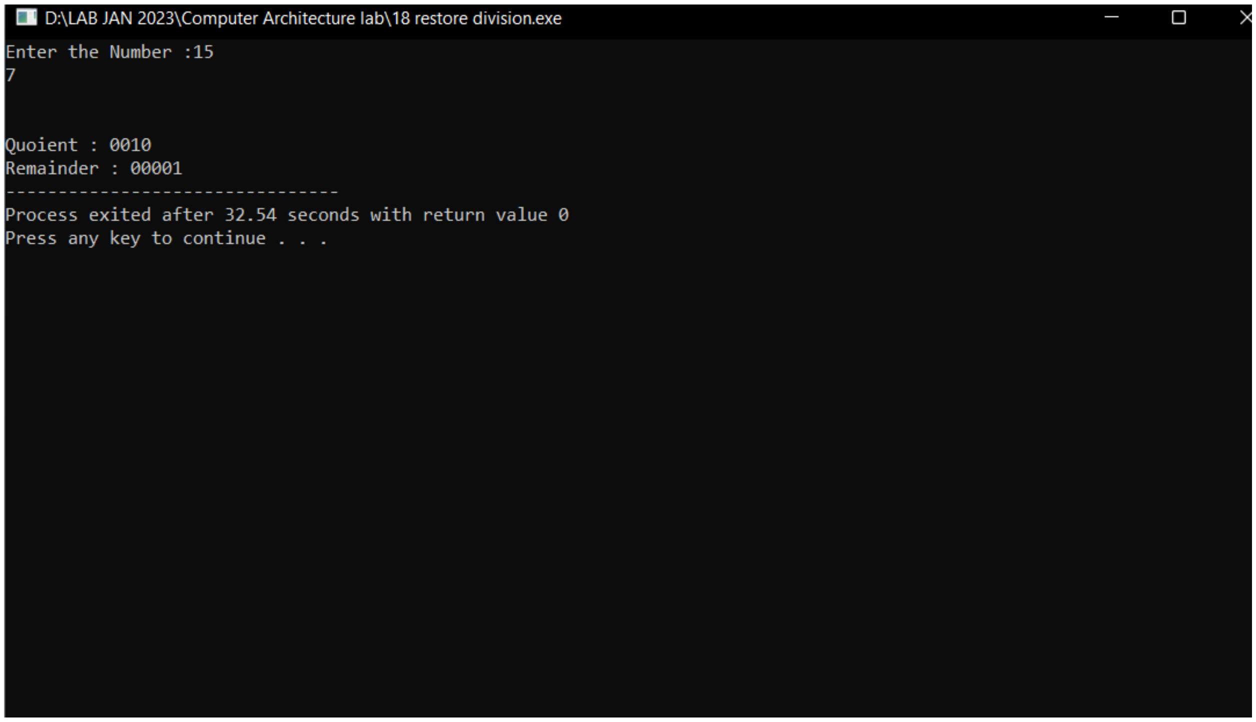
PROGRAM:

```
#include<stdlib.h>
#include<stdio.h>
int acum[100]={0} ;
void add(int acum[],int b[],int n);
int q[100],b[100];
int main()
{
int x,y;
printf("Enter the Number :");
scanf("%d%d",&x,&y);
int i=0;
while(x>0||y>0)
{
if(x>0)
{
q[i]=x%2;
x=x/2;
}
else
{
q[i]=0;
}
if(y>0)
{
b[i]=y%2;
y=y/2;
}
else
{
b[i]=0;
}
i++;
}
int n=i;
int bc[50];
printf("\n");
for(i=0;i<n;i++)
{
if(b[i]==0)
{
bc[i]=1;
}
```

```
else
{
bc[i]=0;
}
}
bc[n]=1;
for(i=0;i<=n;i++)
{
if(bc[i]==0)
{
bc[i]=1;
i=n+2;
}
else
{
bc[i]=0;
}
}
int l;
b[n]=0;
int k=n;
int n1=n+n-1;
int j,mi=n-1;
for(i=n;i!=0;i--)
{
for(j=n;j>0;j--)
{
acum[j]=acum[j-1];
}
acum[0]=q[n-1];
for(j=n-1;j>0;j--)
{
q[j]=q[j-1];
}
add(acum,bc,n+1);
if(acum[n]==1)
{
q[0]=0;
add(acum,b,n+1);
}
else
{
q[0]=1;
}
```

```
}
printf("\nQuoient : ");
for( l=n-1;l>=0;l--)
{
printf("%d",q[l]);
}
printf("\nRemainder : ");
for( l=n;l>=0;l--)
{
printf("%d",acum[l]);
}
return 0;
}
void add(int acum[],int bo[],int n)
{
int i=0,temp=0,sum=0;
for(i=0;i<n;i++)
{
sum=0;
sum=acum[i]+bo[i]+temp;
if(sum==0)
{
acum[i]=0;
temp=0;
}
else if (sum==2)
{
acum[i]=0;
temp=1;
}
else if(sum==1)
{
acum[i]=1;
temp=0;
}
else if(sum==3)
{
acum[i]=1;
temp=1;
}
}
}
```

OUTPUT:



D:\LAB JAN 2023\Computer Architecture lab\18 restore division.exe

```
Enter the Number :15
7


Quoient : 0010
Remainder : 00001
--------------------------------
Process exited after 32.54 seconds with return value 0
Press any key to continue . . .
```

RESULT:

Thus the Program for Restoring Division was executed Successfully.