Restoring division of two numbers

```c
#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 and Divisor :");

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;
}
}
```

<global>   main() : int

Start here × | 3 STAGE PIPELINING OR.c ×

```c
1    #include<stdlib.h>
2    #include<stdio.h>
3    int acum[100]={0}    ;
4    void add(int acum[],int b[],int n);
5    int q[100],b[100];
6    int main()
7    {
8    int x,y;
9    printf("Enter the Number and Divisor :");
10   scanf("%d%d",&x,&y);
11   int i=0;
12   while(x>0||y>0)
13   {
14   if(x>0)
15   {
16   q[i]=x%2;
17   x=x/2;
18   }
19   else
20   {
21   q[i]=0;
22   }
23   if(y>0)
```

```
"C:\Users\Admin\Documents\3 STAGE PIPELINING OR.exe"       —   □   ×

Enter the Number and Divisor :20
3


Quoient   : 00110
Remainder : 000010
Process returned 0 (0x0)    execution time : 6.090 s
Press any key to continue.
```