**INTEGER RESTORING DIVISION:**

**EXP NO:33**

**AIM:**To write a C program to implement integer restoring division.

**APPARATUS:** DEV C++

**ALGORITHM:**

**1)Step-1:** First the registers are initialized with corresponding values (Q = Dividend, M= Divisor, A = 0, n = number of bits in dividend)

**2)Step-2:** Then the content of register A and Q is shifted left as if they are a single unit

**3)Step-3:** Then content of register M is subtracted from A and result is stored in A

**4)Step-4:** Then the most significant bit of the A is checked if it is 0 the least significant bit of Q is set to 1 otherwise if it is 1 the least significant bit of Q is set to 0 and value of register A is restored i.e., the value of A before the subtraction with M

**5)Step-5:** The value of counter n is decremented

**6)Step-6:** If the value of n becomes zero, we get of the loop otherwise we repeat from step 2

**7)Step-7:** Finally, the register Q contain the quotient and A contain remainder

**PROGRAM:**#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;

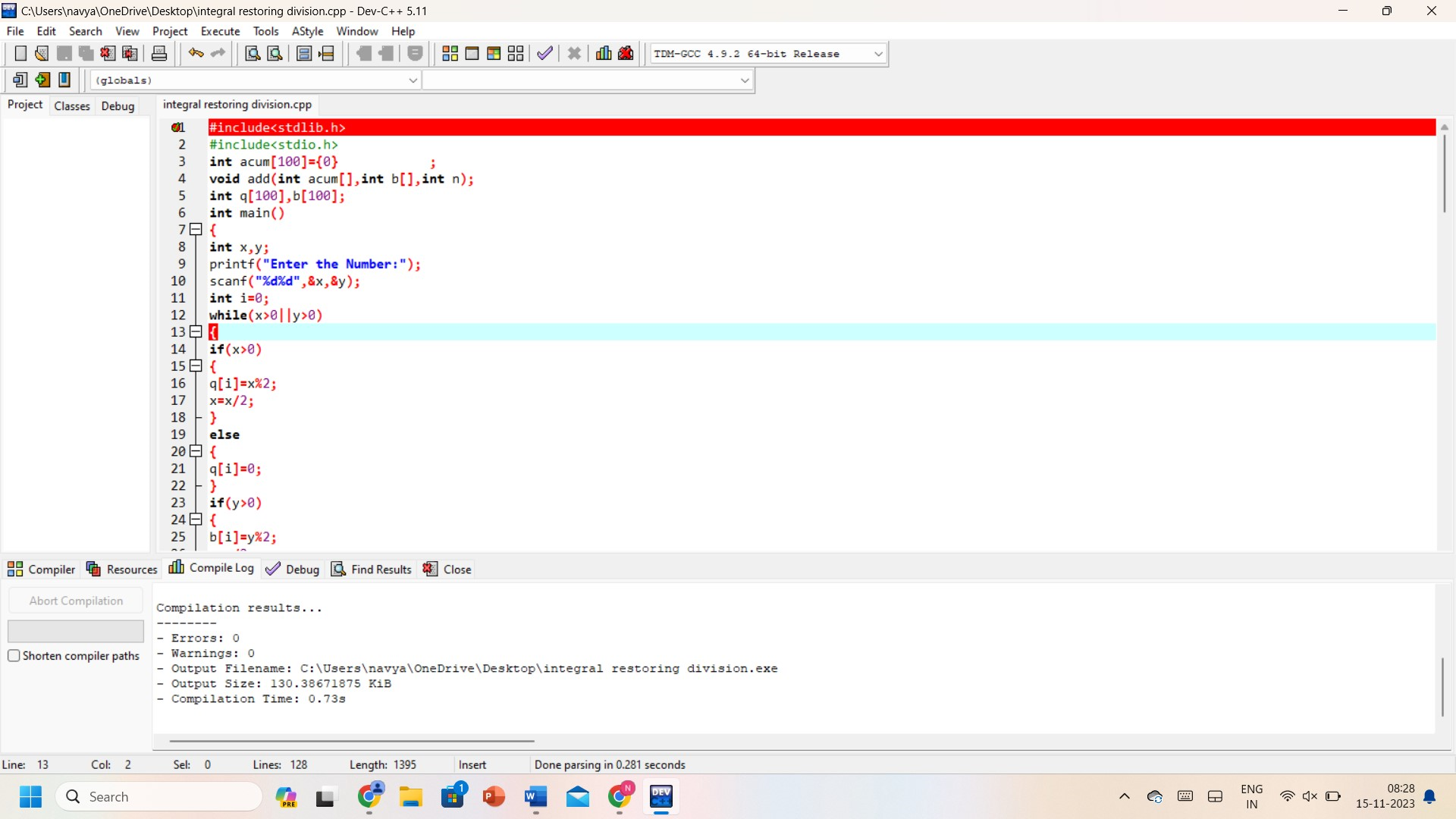
temp=1;

}

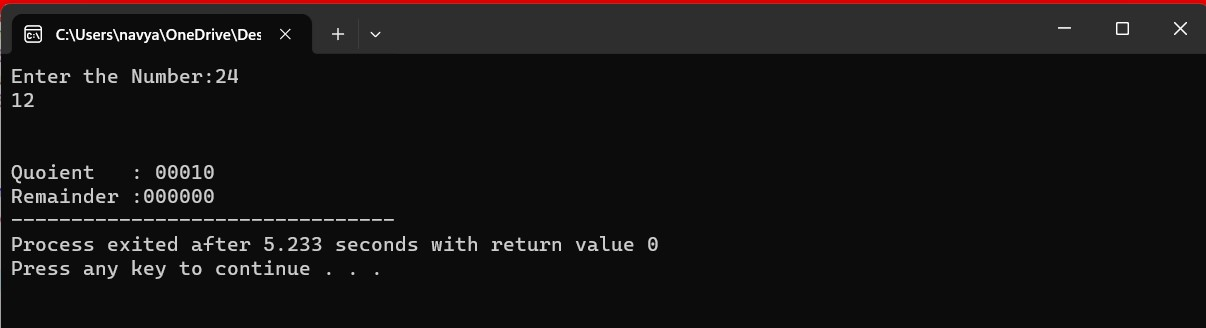
}

}

**INPUT:**



**OUTPUT:**



**RESULT:** Thus, the program was executed successfully using DevC++.