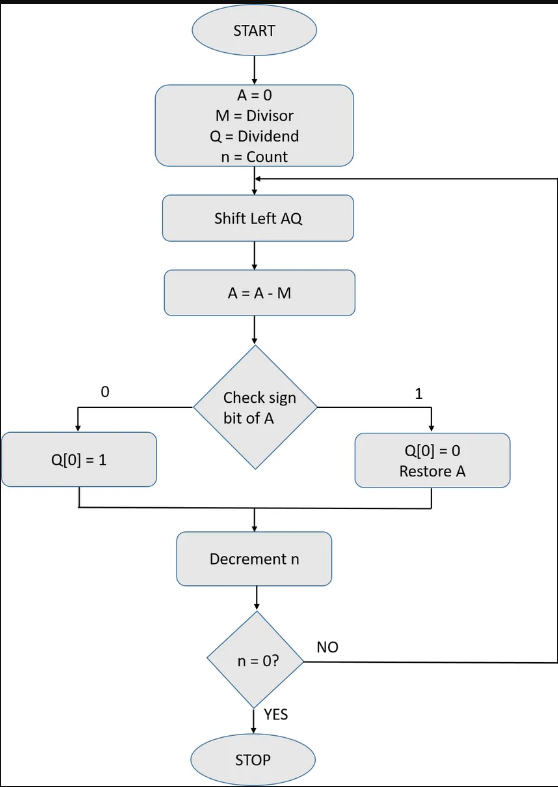
NAME : ADWAIT S PURAO

UID : 2021300101

BATCH : B2

RESTORING DIVISION FLOWCHART:



CODE:

def deci\_to\_bin(n):

    return bin(n).replace("0b", "")

def shift\_left(C,AC,Q):

    C=AC[0]

    temp\_AC=list(AC)

    for i in range(1,len(AC)):

        temp\_AC[i-1]=temp\_AC[i]

    temp\_AC[len(AC)-1]=Q[0]

    AC=''

    AC=AC.join(temp\_AC)

    temp\_Q=list(Q)

    for i in range(1,len(Q)):

        temp\_Q[i-1]=temp\_Q[i]

    temp\_Q[len(Q)-1]='\_'

    Q=''

    Q=Q.join(temp\_Q)

    return(C,AC,Q)

def operation(C,AC,M):

    temp=C+AC

    temp=bin(int(temp,2)+int(M,2))

    temp=temp.replace("0b","")

    # discard the carry while operation is done

    if(len(temp)>len(M)):

        temp=temp[1::]

    return(temp[0],temp[1::])

# Main function

AC=''

C='0'

Q=input("Enter the dividend(Q)  :  ")

Q=deci\_to\_bin(int(Q))

M=input("Enter the divisor(M)   :  ")

M=deci\_to\_bin(int(M))

if(len(Q)>len(M)):

    for i in range(len(Q)):

        AC=AC+'0'

else:

    for i in range(len(M)):

        AC=AC+'0'

print("Initial C value is     : ",C)

print("Initial AC value is    : ",AC)

print("Initial Q value is     : ",Q)

for i in range(len(Q)-len(M)):

    M='0'+M

# adding one bit extra

M='0'+M

print("Value of M is          : ",M)

# two's complement

M\_array=list(M)

for i in range(len(M)):

    if(M[i]=='0'):

        M\_array[i]='1'

    if(M[i]=='1'):

        M\_array[i]='0'

M\_negative=''

M\_negative=M\_negative.join(M\_array)

M\_negative=bin(int(M\_negative,2)+int('1',2))

M\_negative=M\_negative.replace("0b","")

print("Two's complement of M  : ",M\_negative)

print()

print("---------------------------------------------------------------------------------------------------------------------")

print("\t C "," "\*int(len(AC)/2),"AC"," "\*int(len(AC)/2)," "\*int(len(Q)/2),"Q"," "\*int(len(Q)/2),"     Operation done")

print("---------------------------------------------------------------------------------------------------------------------")

print("\t",C,"  ",AC,"  ",Q,"   ","Initial values")

print()

for i in range(len(Q)):

    print("step",(i+1),":  ")

    C,AC,Q=shift\_left(C,AC,Q)

    print("\t",C,"  ",AC,"  ",Q,"   ","After shift left operation")

    C,AC=operation(C,AC,M\_negative)

    print("\t",C,"  ",AC,"  ",Q,"   ","AC\_equals\_AC\_minus\_M operation")

    if(C=='1'):

        temp\_Q=list(Q)

        temp\_Q[len(Q)-1]='0'

        Q=''

        Q=Q.join(temp\_Q)

        C,AC=operation(C,AC,M)

        print("\t",C,"  ",AC,"  ",Q,"   ","AC\_equals\_AC\_plus\_M operation")

    else:

        temp\_Q=list(Q)

        temp\_Q[len(Q)-1]='1'

        Q=''

        Q=Q.join(temp\_Q)

    print()

print('\nFinal values')

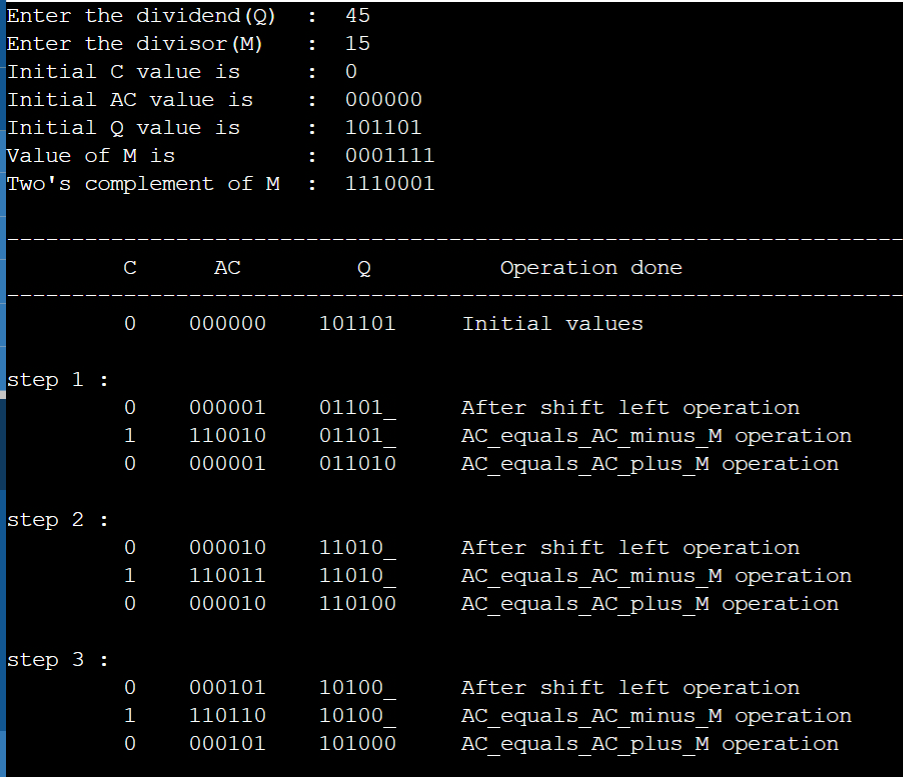
print("\t",C,"  ",AC,"  ",Q)

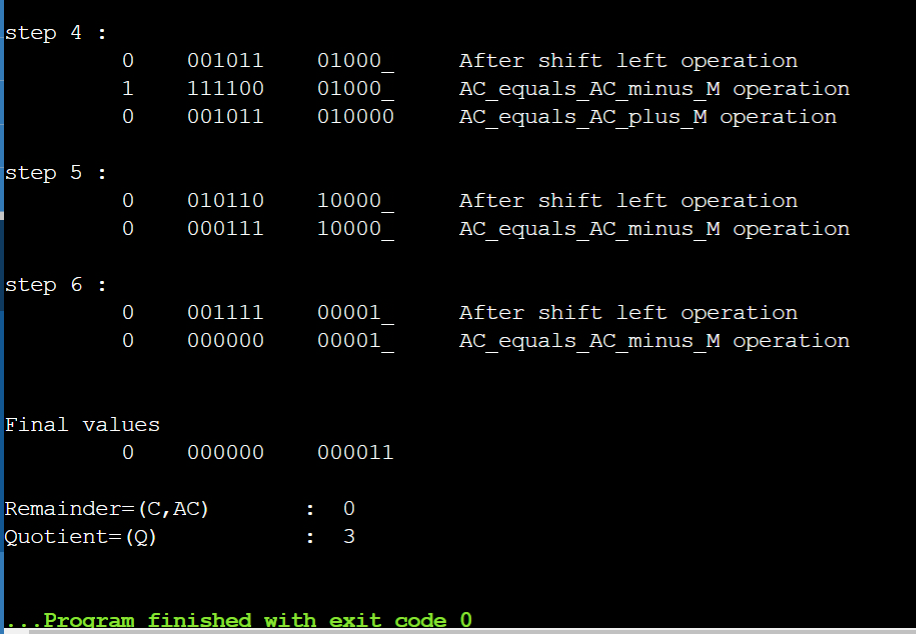
print()

print("Remainder=(C,AC)       : ",int(C+AC,2))

print("Quotient=(Q)           : ",int(Q,2))

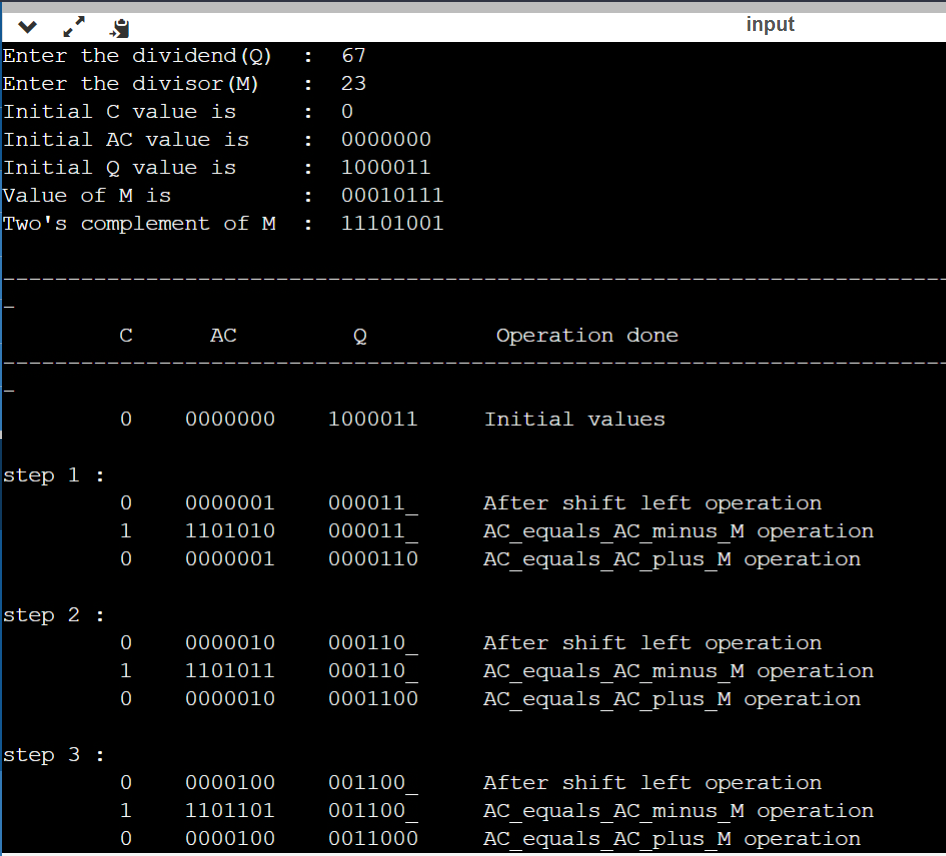
Output:

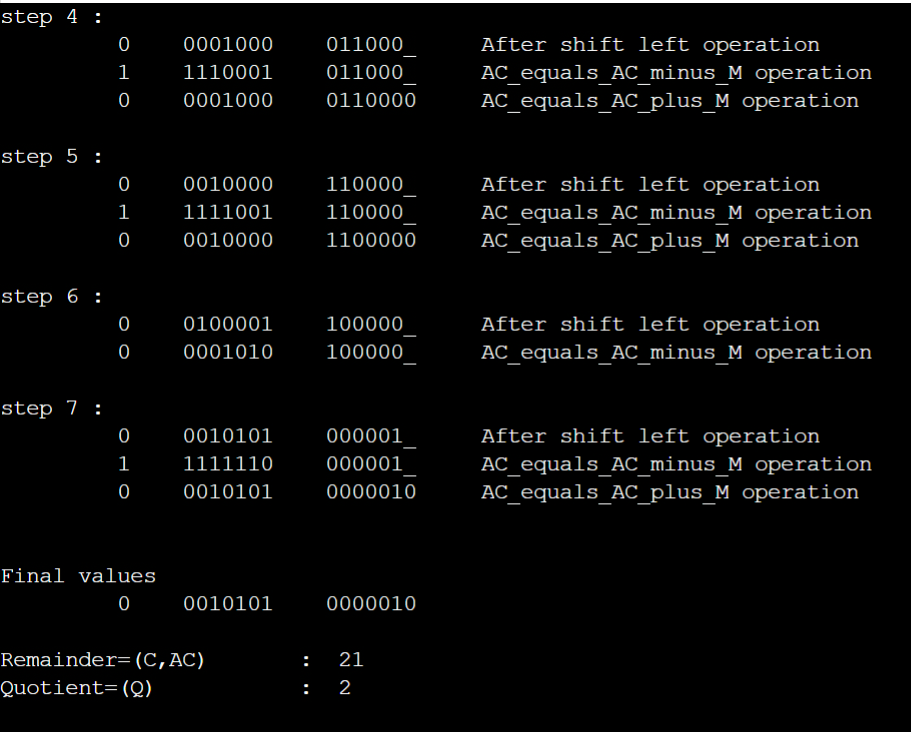




Input 2:

67/23





Conclusion:

We learnt about the restoring division algorithm with the help of flowchart and implemented it with the help of python.