**Experiment No: 01**

**Aim: To implement Booth’s Algorithm for signed and unsigned integer multiplication.**

**Code and Output:**

**Booth’s Algorithm:**

def twosComplement(num):

onesComp=""

for i in num:

if i == "0":

onesComp += "1"

else:

onesComp +="0"

return bin(int(onesComp,2) + int("1",2)).replace('0b',"")

num1 = int(input('Enter number: '))

num2 = int(input('Enter 2nd number: '))

binNum1 = bin(abs(num1)).replace("0b",'')

binNum2 = bin(abs(num2)).replace("0b",'')

if len(binNum1) >= len(binNum2):

maxlen = len(binNum1)

else:

maxlen = len(binNum2)

maxlen +=1

binNum1 = binNum1.zfill(maxlen)

binNum2 = binNum2.zfill(maxlen)

if num2 < 0:

binNum2 = twosComplement(binNum2)

if num1 < 0:

binNum1 = twosComplement(binNum1)

binCompNum1 = twosComplement(binNum1)

binCompNum1 = binCompNum1.zfill(maxlen)

print('Number 1 binary:',binNum1)

print('Number 2 binary:',binNum2)

print('Number 1 complement:',binCompNum1)

count = maxlen

m = binNum1

minusm = binCompNum1

q = binNum2

q1 = '0'

a = "0"

a = a.zfill(maxlen)

rightshift=""

while count > 0:

if q1 == '1' and q[maxlen-1] == '0':

a = bin(int(a,2) + int(m,2)).replace('0b','')

if(len(a) > maxlen):

a = a[1:]

a = a.zfill(maxlen)

elif q1=='0' and q[maxlen-1] == '1':

a = bin(int(a,2) + int(minusm,2)).replace('0b','')

if(len(a) > maxlen):

a = a[1:]

a = a.zfill(maxlen)

merged = a+q+q1

rightshift = merged[0]

for i in range(len(merged)-1):

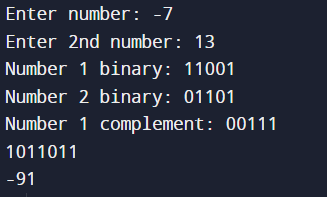
rightshift += merged[i]

a = rightshift[:maxlen]

q = rightshift[maxlen:maxlen\*2]

q1 = rightshift[-1]

count -=1

****ans = a+q

minus = False

if ans[0] == '1':

ans = twosComplement(ans)

minus = True

print(ans)

if minus:

print(int(ans,2) \* -1)

else:

print(int(ans,2))