



A.Y. 2022-2023

Subject: Process Organization and Architecture

SAP ID: 60004220253 – Devansh Mehta

Experiment No: 02

Aim: To implement Restoring and Non-restoring method for integer division.

Code and Output:

Restoring Method:

```
def shift_left(s):
    s=s[1:]
    s=s+"0"
    return (s)
def complement(s):
    d={'0':'1','1':'0'}
    e="".join(d[x] for x in s)
    l=len(e)
    sum=bin(int(e,2)+int("1",2))
    sum=sum[2:]
    return(sum.zfill(l))
def addzerotostring(x):
    x=list(x)
    x[-1]="0"
    return("".join(x))
def addonetostring(x):
    x=list(x)
    x[-1]="1"
    return("".join(x))
q=int(input("Enter the dividend :"))
m=int(input("Enter the divisor :"))
Q=bin(q)
M=bin(m)
# print('q',Q)
Q=Q[2:] #final Q
M=M[2:] #final M
# print('q',Q)
if(len(M)>len(Q)):
    Q=Q.zfill(len(M))
else:
    M=M.zfill(len(Q))
print("Binary Value Of M :",M)
print("Binary Value Of Q :",Q)
l=len(Q)
count=1
Mc=complement(M)
a="0"
```



A.Y. 2022-2023

Subject: Process Organization and Architecture

SAP ID: 60004220253 – Devansh Mehta

```
for i in range(0,l-1):
    a=a+"0"
while(count>0):
    s=a+Q
    value=shift_left(s)
    a=value[0:l]
    Q=value[l:]
    a=bin(int(a,2)+int(Mc,2))
    if(len(a[2:])==l+1):
        a=a[3:]
    else:
        a=a[2:]
    if(a[0]=="1"):
        a=bin(int(a,2)+int(M,2))
        if(len(a[2:])==l+1):
            a=a[3:]
        else:
            a=a[2:]
        Q=addzerotostring(Q)
    else:
        Q=addonetostring(Q)
    count=count-1
print("Remainder in Binary Form :",a)
print("Remainder in Decimal Form :",int(a,2))
print("Quotient in Binary Form :",Q)
print("Quotient in Decimal Form :",int(Q,2))
```

```
Enter the dividend :15
Enter the divisor  :4
Binary Value Of M : 0100
Binary Value Of Q : 1111
Remainder in Binary Form : 0011
Remainder in Decimal Form : 3
Quotient in Binary Form : 0011
Quotient in Decimal Form : 3
```

Non-Restoring Method:

```
def add(A, M):
    carry = 0
    Sum = ""
    for i in range (len(A)-1, -1, -1):
```



A.Y. 2022-2023

Subject: Process Organization and Architecture

SAP ID: 60004220253 – Devansh Mehta

```
temp = int(A[i]) + int(M[i]) + carry
if (temp>1):
    Sum += str(temp % 2)
    carry = 1
else:
    Sum += str(temp)
    carry = 0
return Sum[::-1]
```

```
def compliment(m):
    M = ""
    for i in range (0, len(m)):
        M += str((int(m[i]) + 1) % 2)
    M = add(M, '0001')
    return M
```

```
def nonRestoringDivision(Q, M, A):
    count = len(M)
    comp_M = compliment(M)
    flag = 'successful'
```

```
print ('Initial Values: A:', A,
      ' Q:', Q, ' M:', M)
while (count):
    print ("\nstep:", len(M)-count + 1,
          end = "")
    print (' Left Shift and ', end = "")
    A = A[1:] + Q[0]
    if (flag == 'successful'):
        A = add(A, comp_M)
        print ('subtract: ')
    else:
        A = add(A, M)
        print ('Addition: ')
```

```
print('A:', A, ' Q:',
      Q[1:]+'_', end = "")
if (A[0] == '1'):
    Q = Q[1:] + '0'
    print (' -Unsuccessful')
```



A.Y. 2022-2023

Subject: Process Organization and Architecture

SAP ID: 60004220253 – Devansh Mehta

```
flag = 'unsuccessful'
print ('A:', A, ' Q:', Q,
      '-Addition in next Step')
else:
    Q = Q[1:] + '1'
    print (' Successful')

flag = 'successful'
print ('A:', A, ' Q:', Q,
      '-Subtraction in next step')
count -= 1
print ('\nQuotient(Q):', Q,
      ' Remainder(A):', A)

# Driver code
if __name__ == "__main__":

    dividend = input('Enter 4 digit binary dividend:')
    divisor =input('Enter 4 digit binary divisor:')

    accumulator = '0' * len(dividend)

    nonRestoringDivision(dividend,
                          divisor,
                          accumulator)

Enter 4 digit binary dividend:1010
Enter 4 digit binary divisor:0011
Initial Values: A: 0000 Q: 1010 M: 0011

step: 1 Left Shift and subtract:
A: 1110 Q: 010_ -Unsuccessful
A: 1110 Q: 0100 -Addition in next Step

step: 2 Left Shift and Addition:
A: 1111 Q: 100_ -Unsuccessful
A: 1111 Q: 1000 -Addition in next Step

step: 3 Left Shift and Addition:
A: 0010 Q: 000_ Successful
A: 0010 Q: 0001 -Subtraction in next step

step: 4 Left Shift and subtract:
A: 0001 Q: 001_ Successful
A: 0001 Q: 0011 -Subtraction in next step

Quotient(Q): 0011 Remainder(A): 0001
```