**DIVISON ALGORITHM**

Name: K.Akanksha

Roll no:160119733122

|  |
| --- |
| **Code:**  def add(A, M):      carry = 0      Sum = ''      for i in range (len(A)-1, -1, -1):            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 = ''          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')                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 = '0111'      divisor = '0101'      accumulator = '0' \* len(dividend)      nonRestoringDivision(dividend, divisor,  accumulator) |

**Output:**

Initial Values: A: 0000 Q: 0111 M: 0101

step: 1 Left Shift and subtract:

A: 1011 Q: 111\_ -Unsuccessful

A: 1011 Q: 1110 -Addition in next Step

step: 2 Left Shift and Addition:

A: 1100 Q: 110\_ -Unsuccessful

A: 1100 Q: 1100 -Addition in next Step

step: 3 Left Shift and Addition:

A: 1110 Q: 100\_ -Unsuccessful

A: 1110 Q: 1000 -Addition in next Step

step: 4 Left Shift and Addition:

A: 0010 Q: 000\_ Successful

A: 0010 Q: 0001 -Subtraction in next step

Quotient(Q): 0001 Remainder(A): 0010