Akshala 2018012 Pragya 2018067

Project 2 - Booth's Algorithm

Contents of Submission:-

1. Code
    a. Multiplication and Division
    b. Documentation inside code
2. Report
    a. Multiplication
        i. Algorithm
        ii. Complexity Analysis
        iii. Flow Chart
        iv. Test Cases
    b. Division
        i. Algorithm
        ii. Complexity Analysis
        iii. Flow Chart
        iv. Test Cases

Multiplication

Algorithm used:

1. x and y are the numbers to be multiplied. They are converted to their binary forms. The binary form of -x is also calculated. The number of bits in x and y are represented by x_bits and y_bits.
2. A = x_binary + (y_bits + 1) 0
   S = negative_x_binary + (y_bits + 1) 0
   P = (x_bits) 0 + y_binary + 0
3. Then the following operations is done y_bits times. The last two bits of P are calculated.
    a. If they are 01, then we do P + A and do not take into account the extra bits
    b. If they are 10, then we do P + S and do not take into account the extra bits
    c. If they are 00 or 11, then we take P as it is

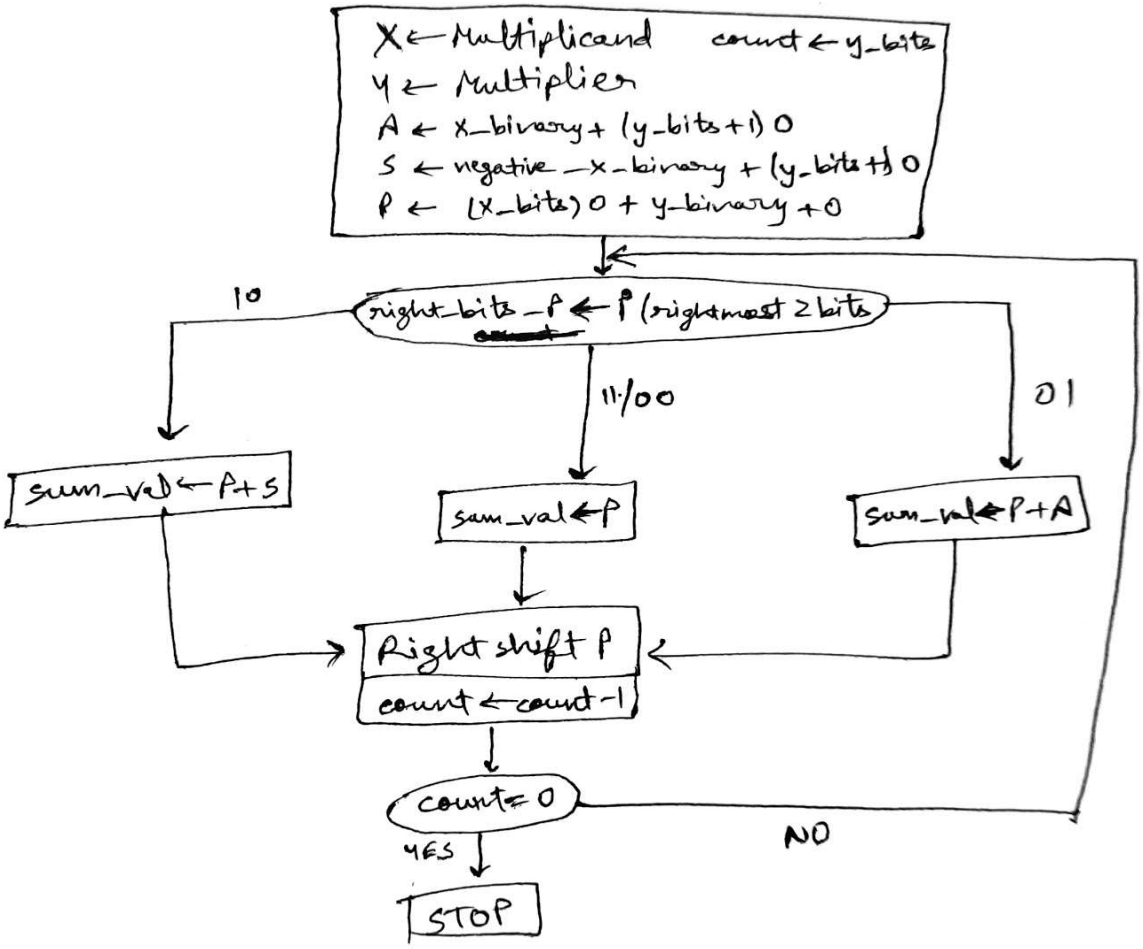
   The above value is shifted to the right by one place. P is now equal to this value.
4. After step 3 is completed, remove the rightmost bit in P. This is the final answer.

Time Complexity: O(n^2)

There is a while loop which runs y_bits time. This complexity can be taken as O(n). Inside the while loop conversions are being done from decimal to binary and vice versa. This complexity is also O(n). Therefore the overall complexity can be taken as O(n^2).

Akshala 2018012 Pragya 2018067

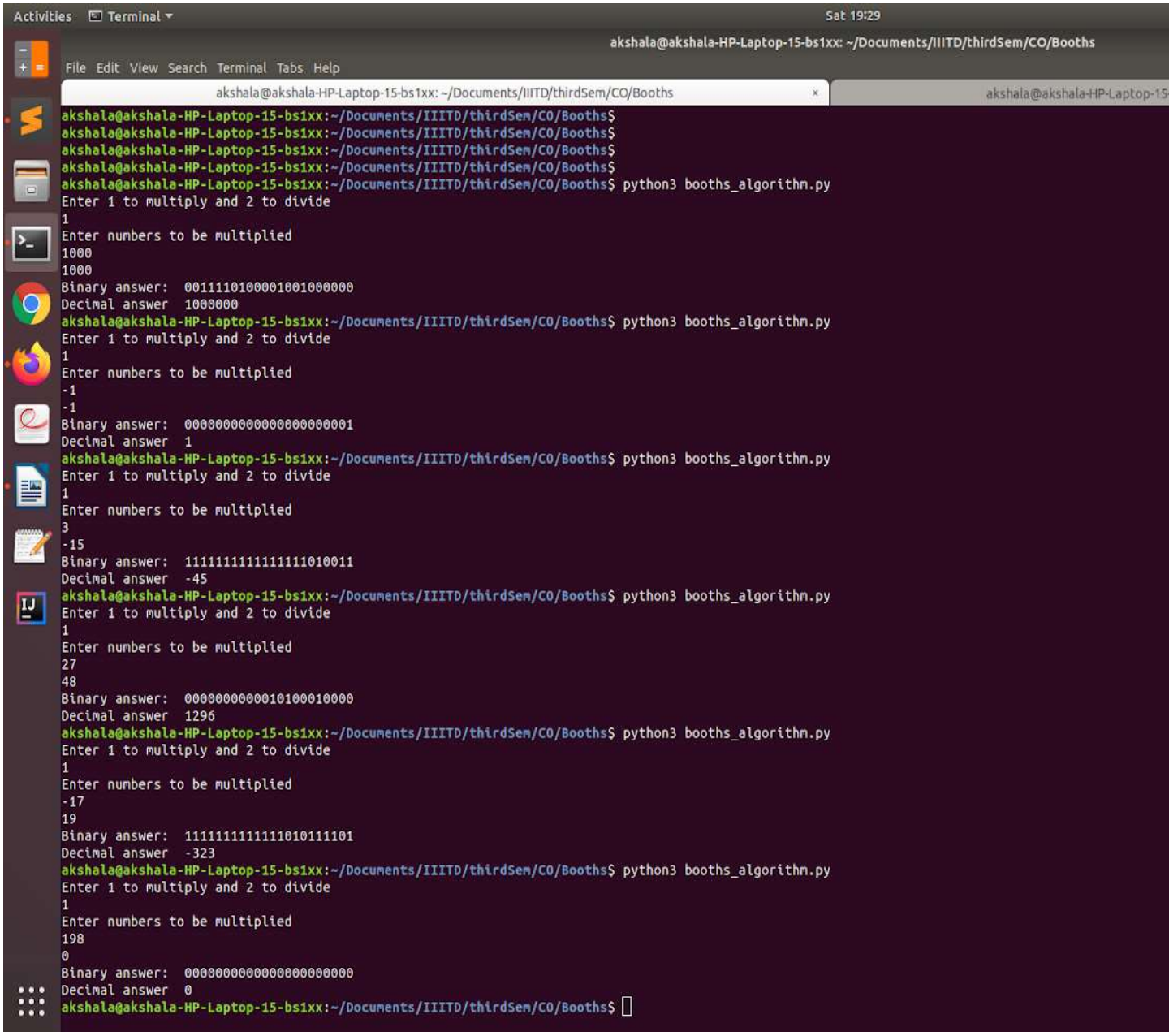
Flow Chart



BOOTH'S ALGORITHM MULTIPLICATION

Test Cases

Akshala 2018012 Pragya 2018067



Division Algorithm - Restoring Division Algorithm

Registers Required:-

1. Div - Stores the value of the divisor.
2. Rem - Stores the value of Remainder.
3. Quo - Stores the value of Quotient
4. Restore - Stores the Restoration Bit

Akshala 2018012 Pragya 2018067

Algorithm used:

1. Initialize the registers with the following values :
    a. Store absolute value of dividend in binary format in Quo. Let the minimum number of bits required to represent dividend be n.
    b. Initialize Div with the value of the absolute value of divisor represented in binary format in n + 1 bits.
    c. Initialize Rem with 0 represented in n + 1 bits.
    d. Initialise the Restore with 0.
2. Repeat the following steps for n number of times :
    a. Shift the bits of Rem to left. MSB of Rem is discarded. LSB of Rem = MSB of Quo.
    b. Subtract the value of Div from Rem. Rem = Rem - Div
    c. If MSB of Rem = 1 :
        i. Restore bit is 0.
        ii. Value of Rem = Value of Rem before step (b)

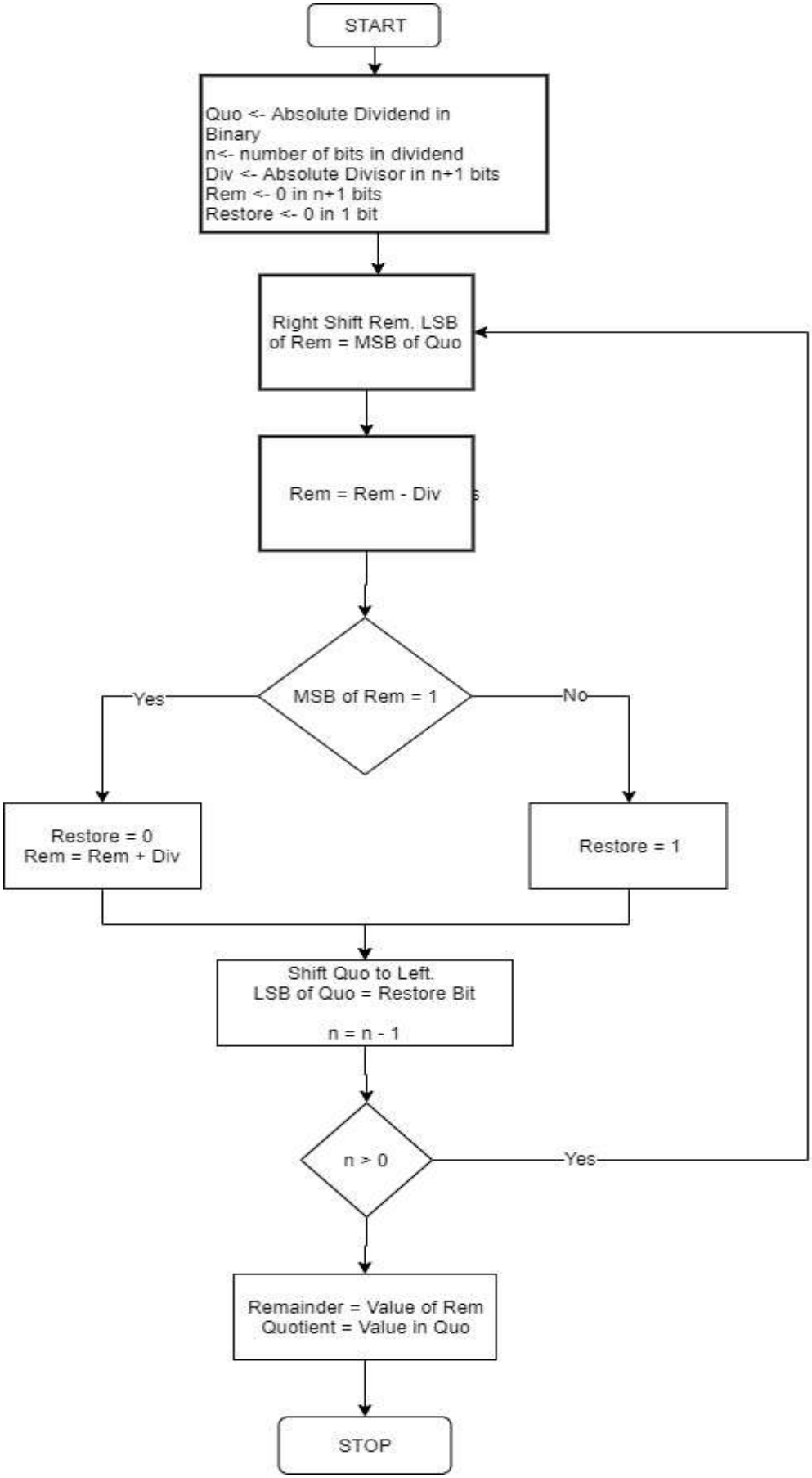
        Else:
        I. Restore bit is 1.
    d. Shift the bits of Quo to left. Discard MSB and LSB = Restore bit.
3. Quotient is the value of Quo, Remainder is the value of Rem.
4. Signed numbers are handled in manner handled in the python programming language:
    a. In the case of negative divisors. Remainder is reported in negative. For example. For (18) / (-5), Quotient = -4 , Remainder = -2 and **not** Quotient = -3 and Remainder = 3

Time Complexity - O(n^2):

‘n’ is the minimum number of bits required to represent dividend. The algorithm loops n times and in each loop (i.e. step 2) Shifting Rem takes O(n), Shifting of Quo takes O(n) times and Subtraction(Div - Rem) takes O(n). Therefore inside each loop O(n) and the number of loops = n give Time Complexity = O(n^2).

Flow Chart :

Akshala 2018012 Pragya 2018067



Akshala 2018012 Pragya 2018067

Test Cases :

1.
```
Enter the value of dividend and enter
600
Enter the value of divisor and enter
200
Quotient is  3
Remainder is  0
```

2.
```
Enter the value of dividend and enter
502
Enter the value of divisor and enter
0
ERROR : Division by 0
```

3.
```
Enter the value of dividend and enter
600
Enter the value of divisor and enter
201
Quotient is  2
Remainder is  198
```

4.
```
Enter the value of dividend and enter
600
Enter the value of divisor and enter
-201
Quotient is  -3
Remainder is  -3
```

5.
```
Enter the value of dividend and enter
-600
Enter the value of divisor and enter
201
Quotient is  -3
Remainder is  3
```

6.
```
Enter the value of dividend and enter
-600
Enter the value of divisor and enter
-201
Quotient is  2
Remainder is  -198
```

7.
```
Enter the value of dividend and enter
0
Enter the value of divisor and enter
502
Quotient is  0
Remainder is  0
```

8.
```
Enter the value of dividend and enter
592
Enter the value of divisor and enter
1
Quotient is  592
Remainder is  0
```

9.

```
Enter the value of dividend and enter
-1050
Enter the value of divisor and enter
25
Quotient is  -42
Remainder is  0
```