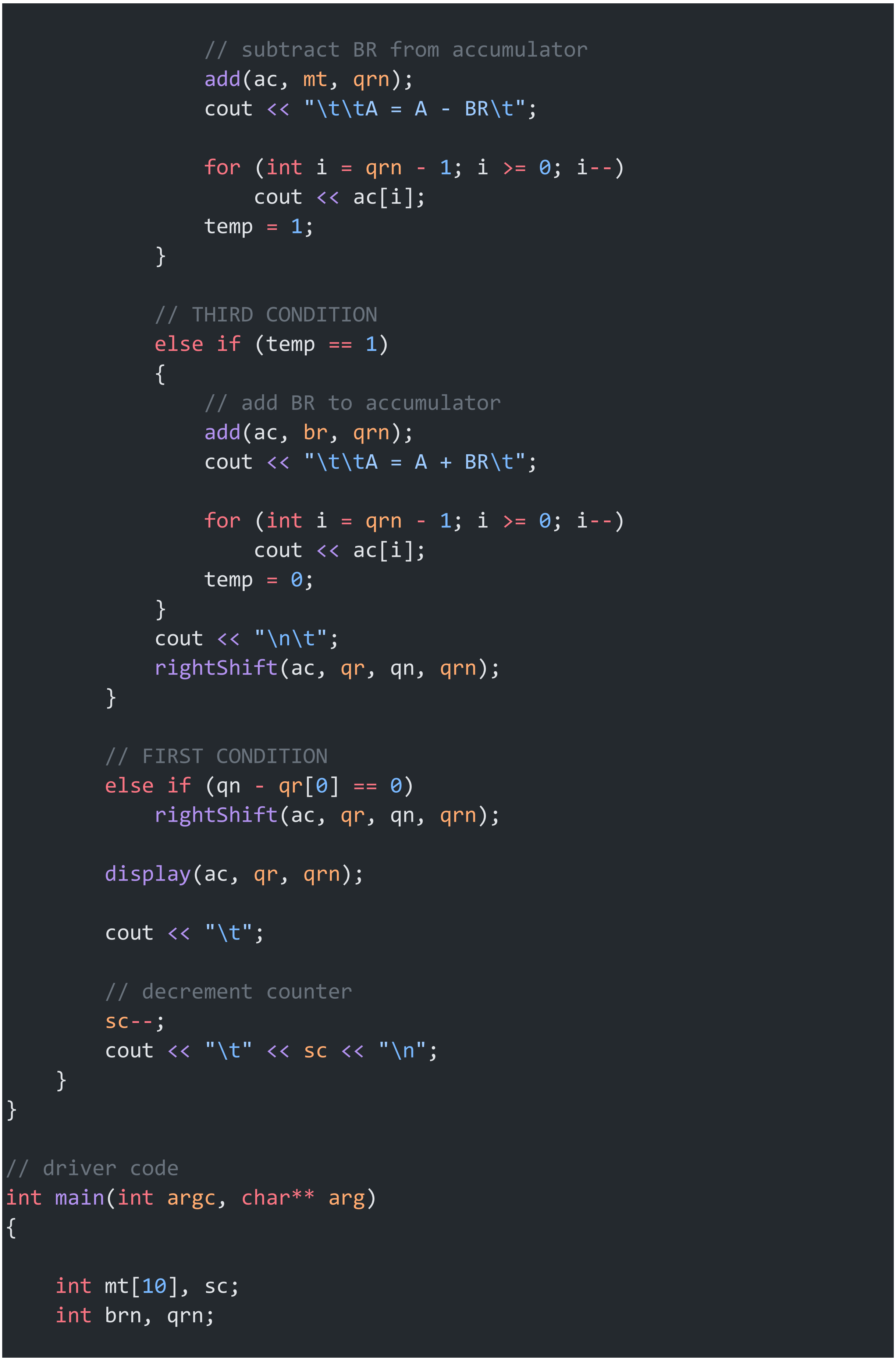
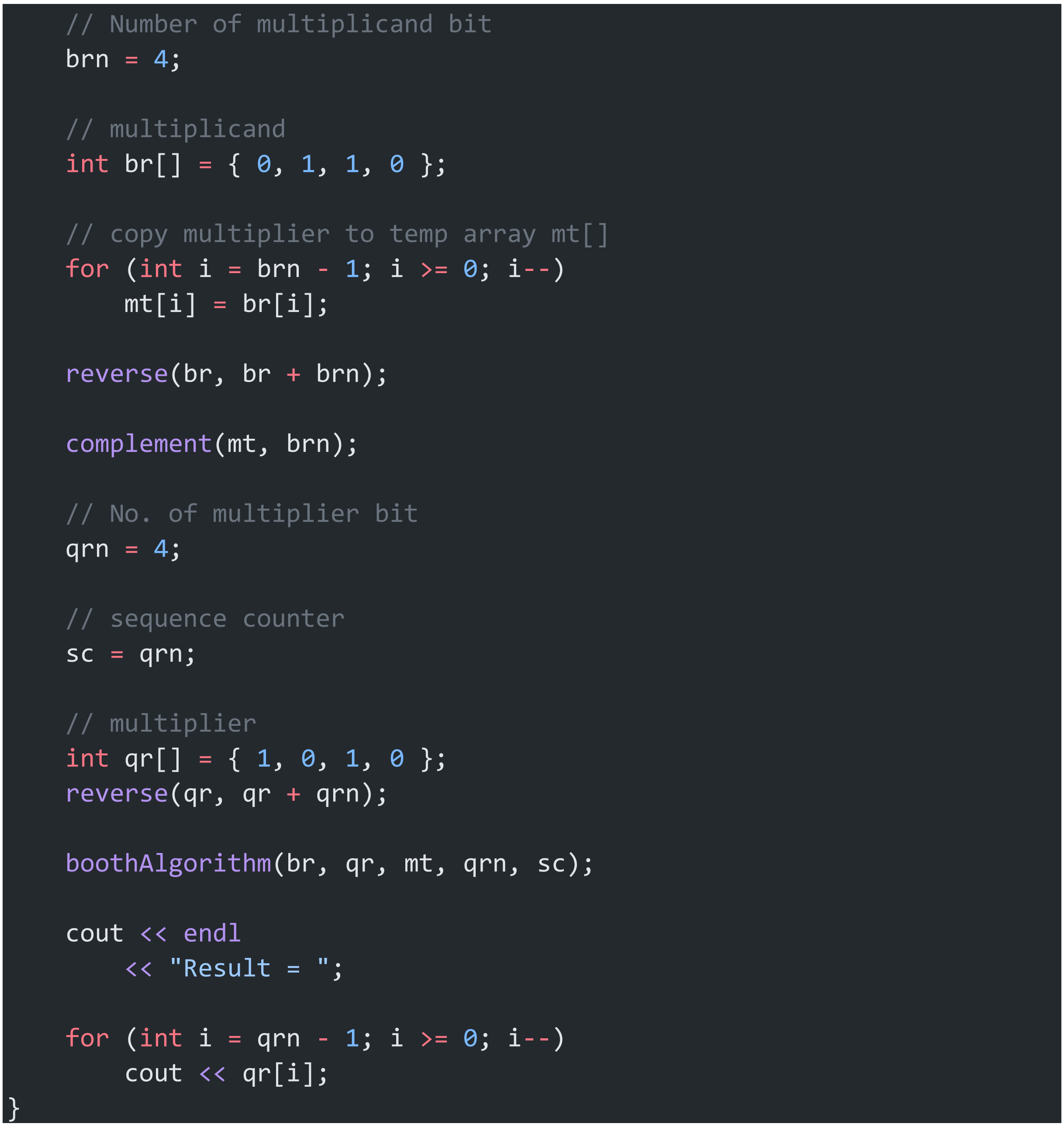
NAME :- Manish Shashikant Jadhav
UID :- 2023301005.
BRANCH :- Comps -B.   **BRANCH:** B.
EXPERIMENT 6: Implement Booth's Multiplication Algorithm.
SUBJECT :- CAO (COMPUTER ARCHITECTURE AND ORGANIZATION)

CODE :-

```
#include <bits/stdc++.h>

using namespace std;

// This function performs adding in accumulator
void add(int ac[], int x[], int qrn)
{
    int i, c = 0;

    for (i = 0; i < qrn; i++) {

        // updating accumulator with A = A + BR
        ac[i] = ac[i] + x[i] + c;

        if (ac[i] > 1) {
            ac[i] = ac[i] % 2;
            c = 1;
        }
        else
            c = 0;
    }
}

// function to find the number's complement
void complement(int a[], int n)
{
    int i;
    int x[8] = {0};
    x[0] = 1;

    for (i = 0; i < n; i++) {
        a[i] = (a[i] + 1) % 2;
    }
    add(a, x, n);
}

// function to perform right shift
void rightShift(int ac[], int qr[], int& qn, int qrn)
{
    int temp, i;
    temp = ac[0];
    qn = qr[0];

    cout << "\t\trightShift\t";

    for (i = 0; i < qrn - 1; i++) {
        ac[i] = ac[i + 1];
        qr[i] = qr[i + 1];
    }
    qr[qrn - 1] = temp;
}

// function to display operations
void display(int ac[], int qr[], int qrn)
{
    int i;

    // accumulator content
    for (i = qrn - 1; i >= 0; i--)
        cout << ac[i];
    cout << "\t";

    // multiplier content
    for (i = qrn - 1; i >= 0; i--)
        cout << qr[i];
}

// Function to implement booth's algo
void boothAlgorithm(int br[], int qr[], int mt[], int qrn, int sc)
{

    int qn = 0, ac[10] = { 0 };
    int temp = 0;
    cout << "qn\tq[n+1]\t\tBR\t\tAC\tQR\t\tsc\n";
    cout << "\t\t\tinitial\t\t";

    display(ac, qr, qrn);
    cout << "\t\t" << sc << "\n";

    while (sc != 0) {
        cout << qr[0] << "\t" << qn;

        // SECOND CONDITION
        if ((qn + qr[0]) == 1)
        {
            if (temp == 0) {
```

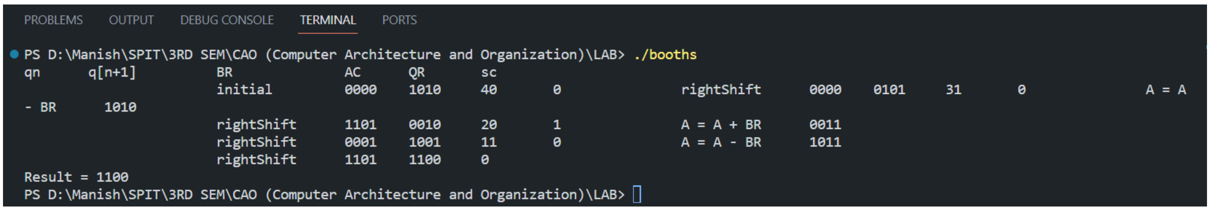


```
                // subtract BR from accumulator
                add(ac, mt, qrn);
                cout << "\t\tA = A - BR\t";

                for (int i = qrn - 1; i >= 0; i--)
                    cout << ac[i];
                temp = 1;
            }

            // THIRD CONDITION
            else if (temp == 1)
            {
                // add BR to accumulator
                add(ac, br, qrn);
                cout << "\t\tA = A + BR\t";

                for (int i = qrn - 1; i >= 0; i--)
                    cout << ac[i];
                temp = 0;
            }
            cout << "\n\t";
            rightShift(ac, qr, qn, qrn);
        }

        // FIRST CONDITION
        else if (qn - qr[0] == 0)
            rightShift(ac, qr, qn, qrn);

        display(ac, qr, qrn);

        cout << "\t";

        // decrement counter
        sc--;
        cout << "\t" << sc << "\n";
    }
}

// driver code
int main(int argc, char** arg)
{

    int mt[10], sc;
    int brn, qrn;
```



```
    // Number of multiplicand bit
    brn = 4;

    // multiplicand
    int br[] = { 0, 1, 1, 0 };

    // copy multiplier to temp array mt[]
    for (int i = brn - 1; i >= 0; i--)
        mt[i] = br[i];

    reverse(br, br + brn);

    complement(mt, brn);

    // No. of multiplier bit
    qrn = 4;

    // sequence counter
    sc = qrn;

    // multiplier
    int qr[] = { 1, 0, 1, 0 };
    reverse(qr, qr + qrn);

    boothAlgorithm(br, qr, mt, qrn, sc);

    cout << endl
        << "Result = ";

    for (int i = qrn - 1; i >= 0; i--)
        cout << qr[i];
}
```

OUTPUT :-



```
PS D:\Manish\SPIT\3RD SEM\CAO (Computer Architecture and Organization)\LAB> ./booths
qn      q[n+1]          BR              AC      QR       sc
                        initial         0000    1010     40       0               rightShift      0000    0101     31       0               A = A
- BR      1010
                        rightShift      1101    0010     20       1               A = A + BR      0011
                        rightShift      0001    1001     11       0               A = A - BR      1011
                        rightShift      1101    1100     0
Result = 1100
PS D:\Manish\SPIT\3RD SEM\CAO (Computer Architecture and Organization)\LAB> 
```