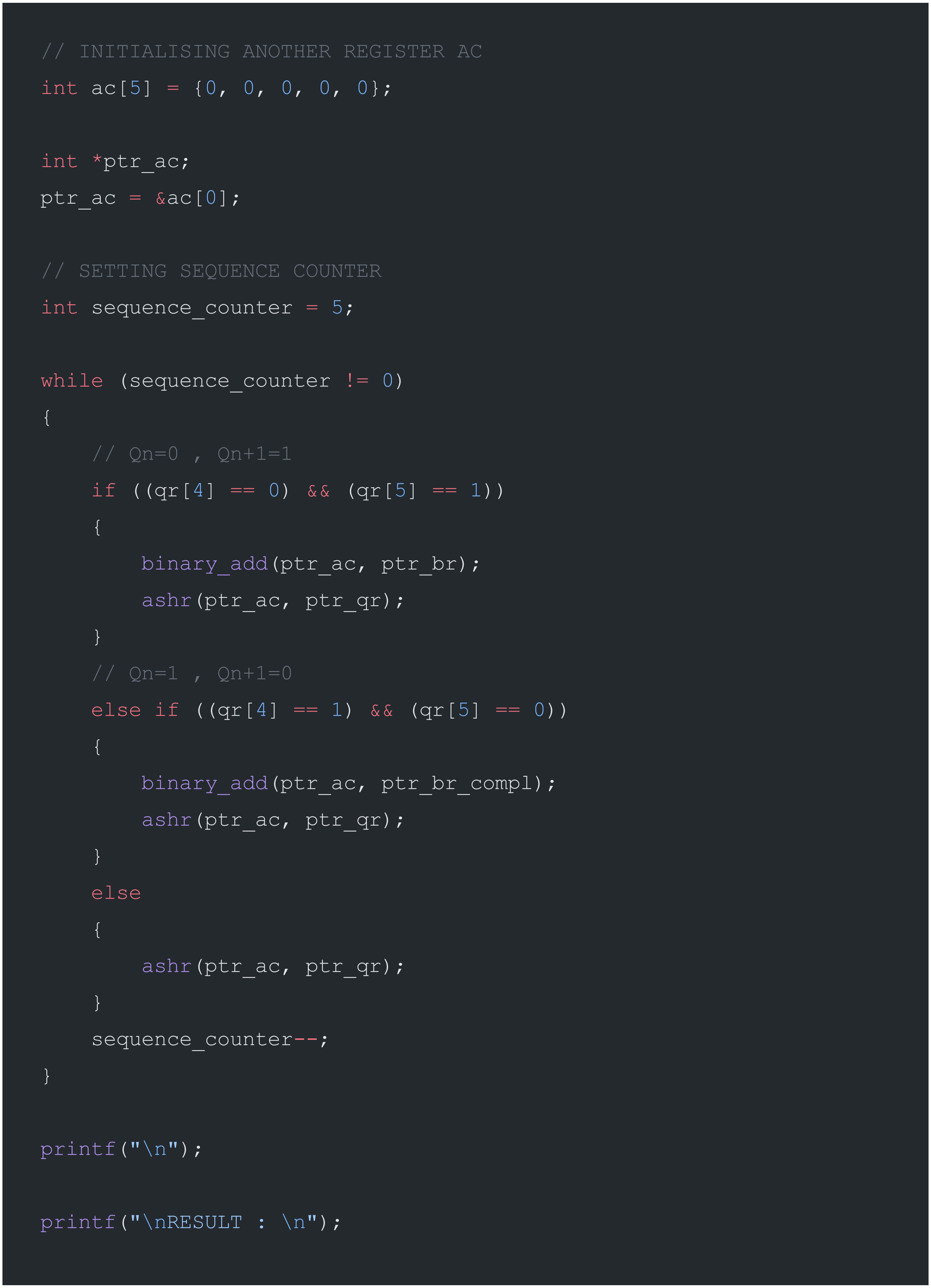
CO ASSIGNMENT-7

-RUSHIL JARIWALA (U21CS001)

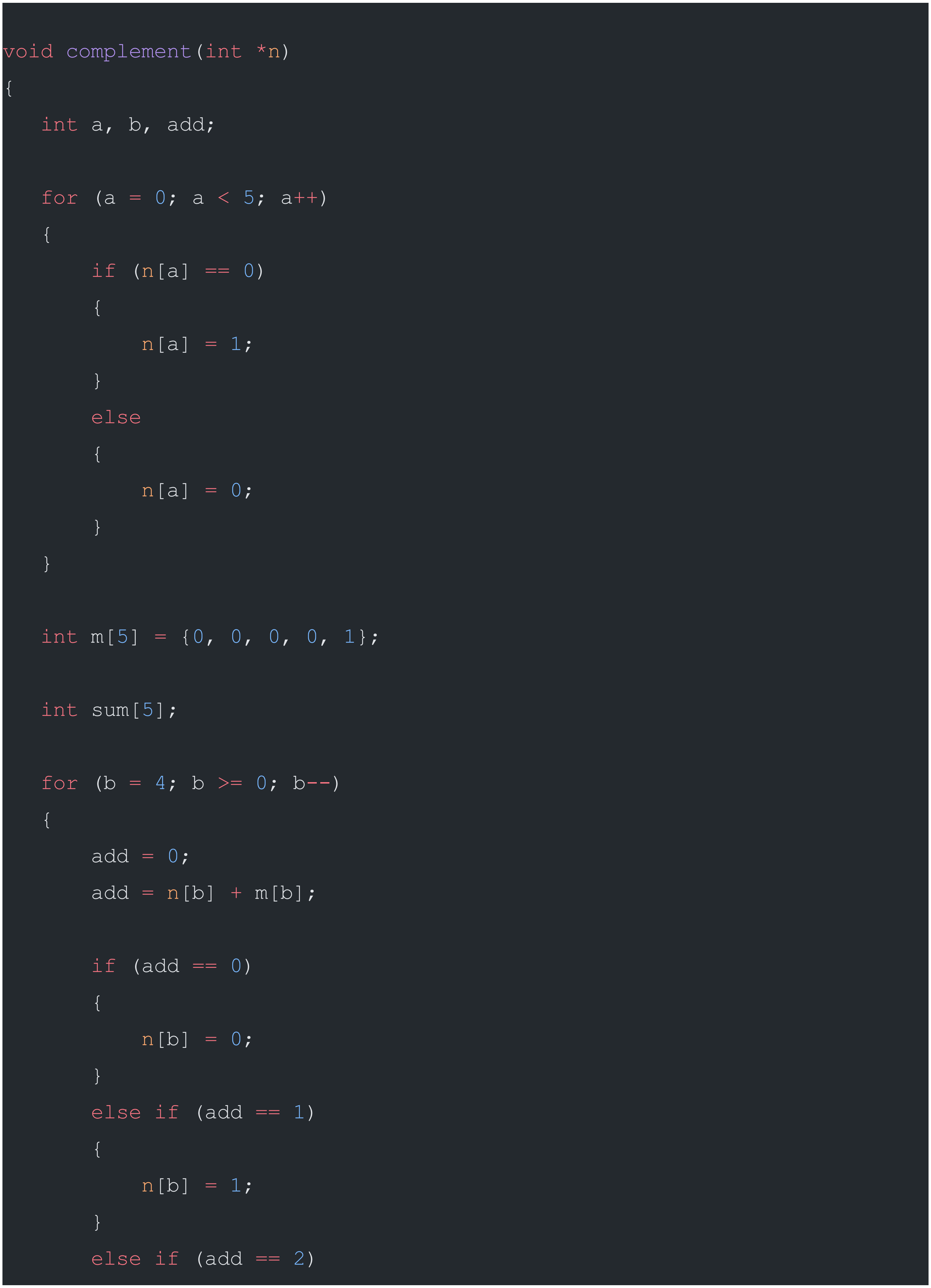
Booth's Algorithm Implementation

```c
#include <stdio.h>
#include <math.h>

void complement(int *);

void binary_add(int *, int *);

void ashr(int *, int *);

int convert_to_decimal(int *);

void complement_result(int *);

int main()
{
    int n1, n2, i, j, l, m;

    printf("\nBOOTH's ALGORITHM \n");
    printf("\nENTER BR as MULTIPLICAND: ");
    scanf("%d", &n1);
    printf("\nENTER QR as MULTIPLIER: ");
    scanf("%d", &n2);

    int br[5];
    int qr[6];

    // MULTIPLICAND IN BR
    for (i = 4; i >= 0; i--)
    {
        br[i] = n1 % 10;
    n1 = n1 / 10;
}

int *ptr_br;
ptr_br = &br[0];

// MULTIPLIER IN QR
for (j = 4; j >= 0; j--)
{
    qr[j] = n2 % 10;
    n2 = n2 / 10;
}
qr[5] = 0;

int *ptr_qr;
ptr_qr = &qr[0];

// COMPLEMENT OF BR
int br_compl[5];

for (l = 0; l < 5; l++)
{
    br_compl[l] = br[l];
}

int *ptr_br_compl;
ptr_br_compl = &br_compl[0];

complement(ptr_br_compl);

/*PRINTING COMPLEMENT
for(l=0;l<5;l++)
{
    printf("%d",br_compl[l]);
}*/
```

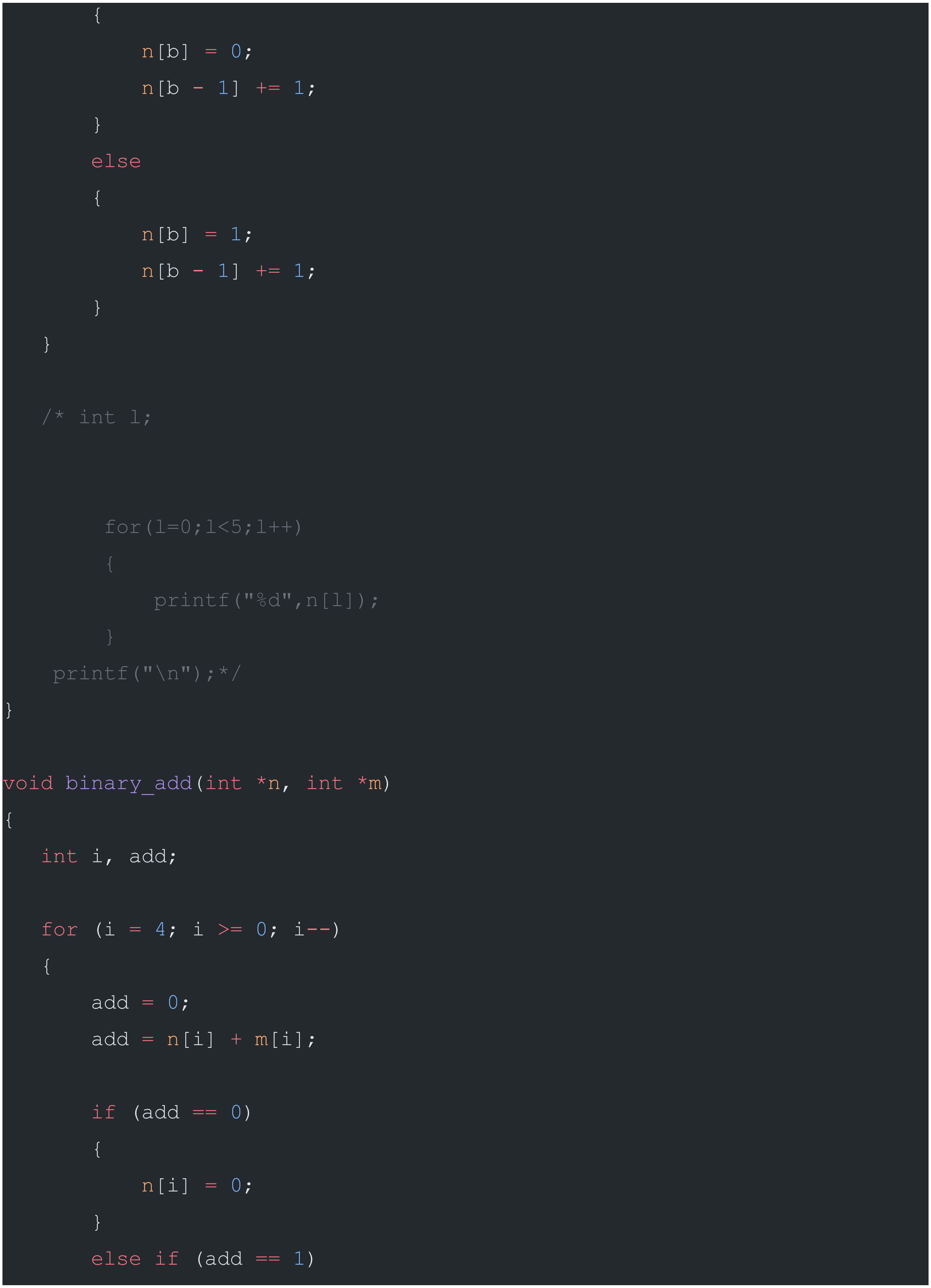


```
// INITIALISING ANOTHER REGISTER AC
int ac[5] = {0, 0, 0, 0, 0};

int *ptr_ac;
ptr_ac = &ac[0];

// SETTING SEQUENCE COUNTER
int sequence_counter = 5;

while (sequence_counter != 0)
{
    // Qn=0 , Qn+1=1
    if ((qr[4] == 0) && (qr[5] == 1))
    {
        binary_add(ptr_ac, ptr_br);
        ashr(ptr_ac, ptr_qr);
    }
    // Qn=1 , Qn+1=0
    else if ((qr[4] == 1) && (qr[5] == 0))
    {
        binary_add(ptr_ac, ptr_br_compl);
        ashr(ptr_ac, ptr_qr);
    }
    else
    {
        ashr(ptr_ac, ptr_qr);
    }
    sequence_counter--;
}

printf("\n");

printf("\nRESULT : \n");
```

```
    int display_arr[10];

    int *ptr;
    ptr = &display_arr[0];

    for (m = 0; m < 10; m++)
    {
        if (m < 5)
        {
            printf("%d", ac[m]);
            display_arr[m] = ac[m];
        }
        else
        {
            printf("%d", qr[m - 5]);
            display_arr[m] = qr[m - 5];
        }
    }

    int result;

    if (display_arr[0] == 0)
    {
        result = convert_to_decimal(ptr);
        printf("\n%d", result);
    }
    else
    {
        complement_result(ptr);
        result = convert_to_decimal(ptr);
        printf("\n - %d", result);
    }

    return 0;
}
```

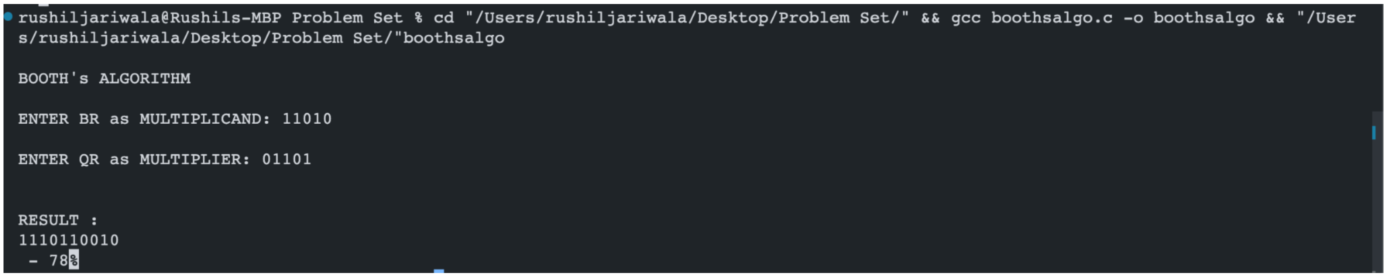


```
void complement(int *n)
{
    int a, b, add;

    for (a = 0; a < 5; a++)
    {
        if (n[a] == 0)
        {
            n[a] = 1;
        }
        else
        {
            n[a] = 0;
        }
    }

    int m[5] = {0, 0, 0, 0, 1};

    int sum[5];

    for (b = 4; b >= 0; b--)
    {
        add = 0;
        add = n[b] + m[b];

        if (add == 0)
        {
            n[b] = 0;
        }
        else if (add == 1)
        {
            n[b] = 1;
        }
        else if (add == 2)
```



```
        {
            n[b] = 0;
            n[b - 1] += 1;
        }
        else
        {
            n[b] = 1;
            n[b - 1] += 1;
        }
    }

    /* int l;


         for(l=0;l<5;l++)
         {
             printf("%d",n[l]);
         }
     printf("\n");*/
}

void binary_add(int *n, int *m)
{
    int i, add;

    for (i = 4; i >= 0; i--)
    {
        add = 0;
        add = n[i] + m[i];

        if (add == 0)
        {
            n[i] = 0;
        }
        else if (add == 1)
        {
            n[i] = 1;
        }
        else if (add == 2)
        {
            n[i] = 0;
            n[i - 1] += 1;
        }
        else
        {
            n[i] = 1;
            n[i - 1] += 1;
        }
    }
}

void ashr(int *n, int *m)
{
    int temp1, i, j;

    temp1 = n[4];
    for (i = 4; i > 0; i--)
    {
        n[i] = n[i - 1];
    }

    for (j = 5; j > 0; j--)
    {
        m[j] = m[j - 1];
    }
    m[0] = temp1;
}

int convert_to_decimal(int *n)
{
    int i, exponent, sum;

    sum = 0;

    for (i = 9; i >= 0; i--)
    {
        exponent = pow(2, (9 - i));
        sum = sum + (n[i] * exponent);
    }

    return sum;
}

void complement_result(int *n)
{
    int a, b, add;

    for (a = 0; a < 10; a++)
    {
        if (n[a] == 0)
        {
            n[a] = 1;
        }
        else
        {
            n[a] = 0;
        }
    }

    int m[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 1};

    for (b = 9; b >= 0; b--)
    {
        add = 0;
        add = n[b] + m[b];
        if (add == 0)
        {
            n[b] = 0;
        }
        else if (add == 1)
        {
            n[b] = 1;
        }
        else if (add == 2)
        {
            n[b] = 0;
            n[b - 1] += 1;
        }
        else
        {
            n[b] = 1;
            n[b - 1] += 1;
        }
    }
}
```

Output



```
rushiljariwala@Rushils-MBP Problem Set % cd "/Users/rushiljariwala/Desktop/Problem Set/" && gcc boothsalgo.c -o boothsalgo && "/User
s/rushiljariwala/Desktop/Problem Set/"boothsalgo

BOOTH's ALGORITHM

ENTER BR as MULTIPLICAND: 11010

ENTER QR as MULTIPLIER: 01101


RESULT :
1110110010
 - 78%
```