

Assignment 9

Q. Write a C code to Implement Restoring Division Algorithm of two signed magnitude numbers.

Input: Two binary numbers

Output: Quotient & Remainder In binary Format

Code -

```
#include<stdio.h>
#include <stdlib.h>
#include <string.h>

//display
void display(int *a, int n)
{
    int i = 0;
    while(i<n)
    {
        printf("%d ", a[i]);
        i++;
    }
}

//complement
void complement(int *a, int n)
{
    int i = 0;
    //1's complement
    while(i<n)
    {
        a[i] = 1 - a[i];
        i++;
    }
    //2's complement
    for (int i = n-1; i >= 0; i--)
    {
        if(a[i] == 0)
        {
            a[i] = 1;
            break;
        }
        else{
            a[i] = 0;
        }
    }
}
```

```

int addition(int *a, int *b, int n)
{
    int Cin = 0;
    for (int i = n-1; i >= 0 ; i--)
    {
        a[i] = a[i] + b[i] + Cin;

        if(a[i] > 1)
        {
            a[i] = a[i]%2;
            Cin = 1;
        }
        else
            Cin = 0;
    }
    return Cin;
}

void shift(int *a, int *q, int *e, int n)
{
    *e = a[0];

    //shift a array
    int i = 0;
    for ( i = 0; i <n-1 ; i++)
    {
        a[i] = a[i+1];
    }
    a[i] = q[0];

    //shift q array

    for (i = 0; i <n-1 ; i++)
    {
        q[i] = q[i+1];
    }
    q[i] = 0;
}

int main()
{
    int n;
    printf("Enter the no. of bits : ");
    scanf("%d", &n);

    //dividend
    int a[20];
    int q[20];

```

```

for (int i = 0; i < n; i++)
{
    q[i] = 0;
}

//divisor
int b[20];

//carry bit
int e = 0;

//input of divisor
printf("Enter the Divisor : ");
for (int i = 0; i < n; i++)
{
    scanf("%d", &b[i]);
}

//input of dividend
printf("Enter the Dividend : ");
for (int i = 0; i < n; i++)
{
    scanf("%d", &a[i]);
}

//sequence counter
int sc = n;

//complement of divisor
int b_comp[20];
memcpy(&b_comp, &b, sizeof(b));
complement(b_comp,n);

//condition for overflow
e = addition(a,b_comp,n);

if(e == 1)
{
    printf("Overflow\n");
}

else{
    //converting to normal form(not a overflow)
    e = addition(a,b,n);

    int faltu;
    while(sc--)
    {
        shift(a,q,&e,n);

        //condition check
    }
}

```

```

        if(e == 0)
        {
            //add the divisor complement
            e = addition(a,b_comp,n);

            //condition check
            if(e == 1)
            {
                //keeping Qn bit as 1
                q[n-1] = 1;
            }
            else{
                //add the divisor
                e = addition(a,b,n);
            }
        }

        else
        {
            faltu = addition(a,b_comp,n);
            q[n-1] = 1;
        }
    }

    printf("Remainder is : ");
    display(a,n);
    printf("\n");

    printf("Quotient is : ");
    display(q,n);
    printf("\n");

    return 0;
}

```

Enter the no. of bits : 5

Enter the Divisor : 1

1

0

1

0

Enter the Dividend : 1

0

0

0

1

Remainder is : 1 1 0 0 0

Quotient is : 1 0 1 0 0