

```
//BOOTH'S ALGORITHM
```

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

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

```
Int a = 0, b = 0, c = 0, a1 = 0, b1 = 0, com[5] = { 1, 0, 0, 0, 0};
```

```
Int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0};
```

```
Int acomp[5] = {0}, bcomp[5] = {0}, pro[5] = {0}, res[5] = {0};
```

```
Void binary(){
```

```
    A1 = fabs(a);
```

```
    B1 = fabs(b);
```

```
    Int r, r2, l, temp;
```

```
    For (l = 0; l < 5; l++){
```

```
        R = a1 % 2;
```

```
        A1 = a1 / 2;
```

```
        R2 = b1 % 2;
```

```
        B1 = b1 / 2;
```

```
        Anum[l] = r;
```

```
        Anumcp[l] = r;
```

```
        Bnum[l] = r2;
```

```
        If(r2 == 0){
```

```
            Bcomp[l] = 1;
```

```
        }
```

```
        If(r == 0){
```

```
            Acomp[l] = 1;
```

```
        }
```

```
    }
```

```
//part for two's complementing
```

```
C = 0;
```

```

For ( l = 0; l < 5; i++){
    Res[i] = com[i]+ bcomp[i] + c;
    If(res[i] >= 2){
        C = 1;
    }
    Else
        C = 0;
    Res[i] = res[i] % 2;
}
For (l = 4; l >= 0; i--){
    Bcomp[i] = res[i];
}
//in case of negative inputs
If (a < 0){
    C = 0;
    For (l = 4; l >= 0; i--){
        Res[i] = 0;
    }
    For ( l = 0; l < 5; i++){
        Res[i] = com[i] + acomp[i] + c;
        If (res[i] >= 2){
            C = 1;
        }
        Else
            C = 0;
        Res[i] = res[i]%2;
    }
    For (l = 4; l >= 0; i--){
        Anum[i] = res[i];
    }
}

```

```

        Anumcp[i] = res[i];
    }

}

If(b < 0){
    For (l = 0; l < 5; i++){
        Temp = bnum[i];
        Bnum[i] = bcomp[i];
        Bcomp[i] = temp;
    }
}

}

Void add(int num[]){
    Int l;
    C = 0;
    For ( l = 0; l < 5; i++){
        Res[i] = pro[i] + num[i] + c;
        If (res[i] >= 2){
            C = 1;
        }
        Else{
            C = 0;
        }
        Res[i] = res[i]%2;
    }
    For (l = 4; l >= 0; i--){
        Pro[i] = res[i];
        Printf("%d",pro[i]);
    }
}

```

```

Printf(":");

For (l = 4; l >= 0; l--){
    Printf("%d", anumcp[l]);
}

}

Void arshift(){//for arithmetic shift right

    Int temp = pro[4], temp2 = pro[0], l;

    For (l = 1; l < 5 ; l++){//shift the MSB of product
        Pro[l-1] = pro[l];
    }

    Pro[4] = temp;

    For (l = 1; l < 5 ; l++){//shift the LSB of product
        Anumcp[l-1] = anumcp[l];
    }

    Anumcp[4] = temp2;

    Printf("\nAR-SHIFT: ");//display together

    For (l = 4; l >= 0; l--){
        Printf("%d",pro[l]);
    }

    Printf(":");

    For(l = 4; l >= 0; l--){
        Printf("%d", anumcp[l]);
    }

}

```

```

Void main(){

    Int l, q = 0;

    Printf("\t\tBOOTH'S MULTIPLICATION ALGORITHM");

    Printf("\nEnter two numbers to multiply: ");

```

```

Printf("\nBoth must be less than 16");
//simulating for two numbers each below 16
Do{
    Printf("\nEnter A: ");
    Scanf("%d",&a);
    Printf("Enter B: ");
    Scanf("%d", &b);
}while(a >=16 || b >=16);

Printf("\nExpected product = %d", a * b);
Binary();
Printf("\n\nBinary Equivalents are: ");
Printf("\nA = ");
For (l = 4; l >= 0; l--){
    Printf("%d", anum[l]);
}
Printf("\nB = ");
For (l = 4; l >= 0; l--){
    Printf("%d", bnum[l]);
}
Printf("\nB' + 1 = ");
For (l = 4; l >= 0; l--){
    Printf("%d", bcomp[l]);
}
Printf("\n\n");
For (l = 0; l < 5; l++){
    If (anum[l] == q){//just shift for 00 or 11
        Printf("\n→");
        Arshift();
    }
}

```

```

        Q = anum[i];
    }
    Else if(anum[i] == 1 && q == 0){//subtract and shift for 10
        Printf("\n→");
        Printf("\nSUB B: ");
        Add(bcomp);//add two's complement to implement subtraction
        Arshift();
        Q = anum[i];
    }
    Else{//add ans shift for 01
        Printf("\n→");
        Printf("\nADD B: ");
        Add(bnum);
        Arshift();
        Q = anum[i];
    }
}

Printf("\nProduct is = ");
For (l = 4; l >= 0; l--){
    Printf("%d", pro[l]);
}
For (l = 4; l >= 0; l--){
    Printf("%d", anumcp[l]);
}

```

}

main.c	Output
<pre>1 //BOOTH'S ALGORITHM 2 #include <stdio.h> 3 #include <math.h> 4 5 int a = 0, b = 0, c = 0, a1 = 0, b1 = 0, com[5] = { 1, 0, 0, 0, 0}; 6 int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0}; 7 int acomp[5] = {0}, bcomp[5] = {0}, pro[5] = {0}, res[5] = {0}; 8 9 void binary(){ 10 a1 = fabs(a); 11 b1 = fabs(b); 12 int r, r2, i, temp; 13 for (i = 0; i < 5; i++){ 14 r = a1 % 2; 15 a1 = a1 / 2; 16 r2 = b1 % 2; 17 b1 = b1 / 2;</pre>	<pre>/tmp/FqkPEhmc5r.o BOOTH'S MULTIPLICATION ALGORITHM Enter two numbers to multiply: Both must be less than 16 Enter A: 3 Enter B: 5 Expected product = 15 Binary Equivalents are: A = 00011 B = 00101 B' + 1 = 11011 --> SUB B: 11011:00011 AR-SHIFT: 11101:10001 --> AR-SHIFT: 11110:11000</pre>

Get Started