

/*C Program to implement Extended-Euclidean Algorithm

Input : Two integers (a,b)

Output : GCD of a and b, coefficients x and y

such that $\text{GCD} = ax + by$

***/**

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

```
int extendedEuclid(int , int , int *, int *);
```

```
int main()
```

```
{
```

```
    int a, b, d, x, y, temp;
```

```
    printf("\n\n I want to find the GCD of two numbers.\n\n");
```

```
    printf(" What are those two numbers?\n\n ");
```

```
    scanf("%d%d",&a,&b);
```

```
    if(a<0) //Treat negative integer just like positive integer
```

```
    a = abs(a);
```

```
    //Either way the answer will be same because we are finding greatest  
    factor and it will be positive only.
```

```
    if(b<0)
```

```
    b = abs(b);
```

```
    if(a < b)
```

```
    {
```

```
        temp =a; //If a is less than b, change their position
```

```
        a=b;
```

```
        b= temp;
```

```
    }
```

C Program on Extended Euclidean Algorithm

```
d = extendedEuclid(a, b, &x, &y);

printf("\n GCD of %d and %d is %d. \n The coefficients are %d and %d.\n\n ",a,b,d,x,y);

return 0;
}

int extendedEuclid(int a, int b, int *x, int *y)
{
    if (a == 0)
    {
        *x = 0;
        *y = 1;
        return b;
    }

    int x1, y1; // To store results of recursive call
    int gcd = extendedEuclid(b%a, a, &x1, &y1);

    *x = y1 - (b/a) * x1;
    *y = x1;

    return gcd;
}
```

Sample input and output

1. When both a and b are +ve integers

```
I want to find the GCD of two numbers.  
What are those two numbers?  
99 78  
GCD of 99 and 78 is 3.  
The coefficients are -11 and 14.  
  
Process returned 0 (0x0)   execution time : 3.770 s  
Press any key to continue.
```

2. When b = 0

```
I want to find the GCD of two numbers.  
What are those two numbers?  
125 0  
GCD of 125 and 0 is 125.  
The coefficients are 1 and 0.  
  
Process returned 0 (0x0)   execution time : 6.906 s  
Press any key to continue.
```