Om Shete - C3 - 2103163

Experiment No 1

Aim: Implementation of Extended Euclidean algorithm.

Theory:

The Euclidean algorithm is a way to find the greatest common divisor of two positive integers. GCD of two numbers is the largest number that divides both of them. A simple way to find GCD is to factorize both numbers and multiply common prime factors.

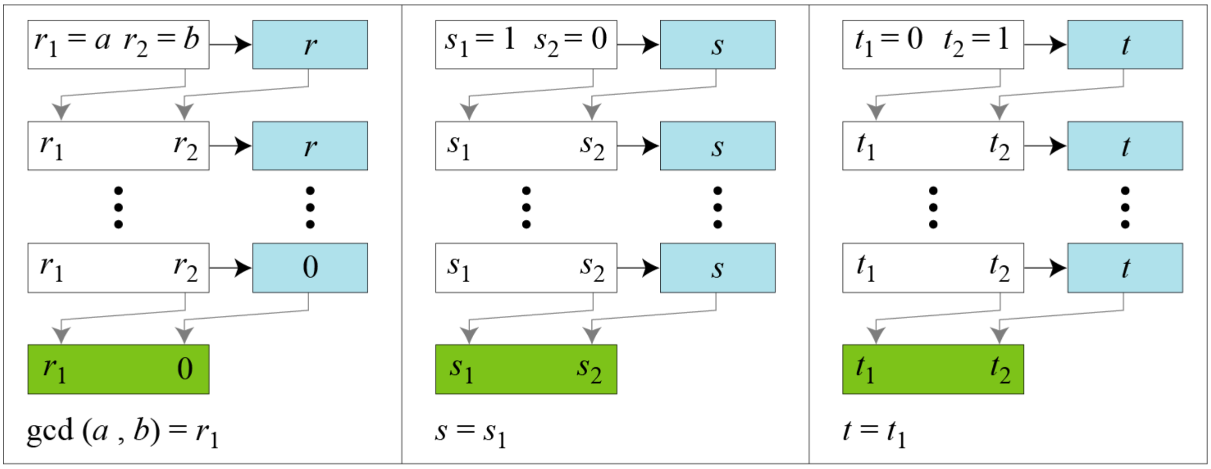
36 = 2 x 2 x 3 x 3
60 = 2 x 2 x 3 x 5

GCD = Multiplication of common factors
= 2 x 2 x 3
= 12

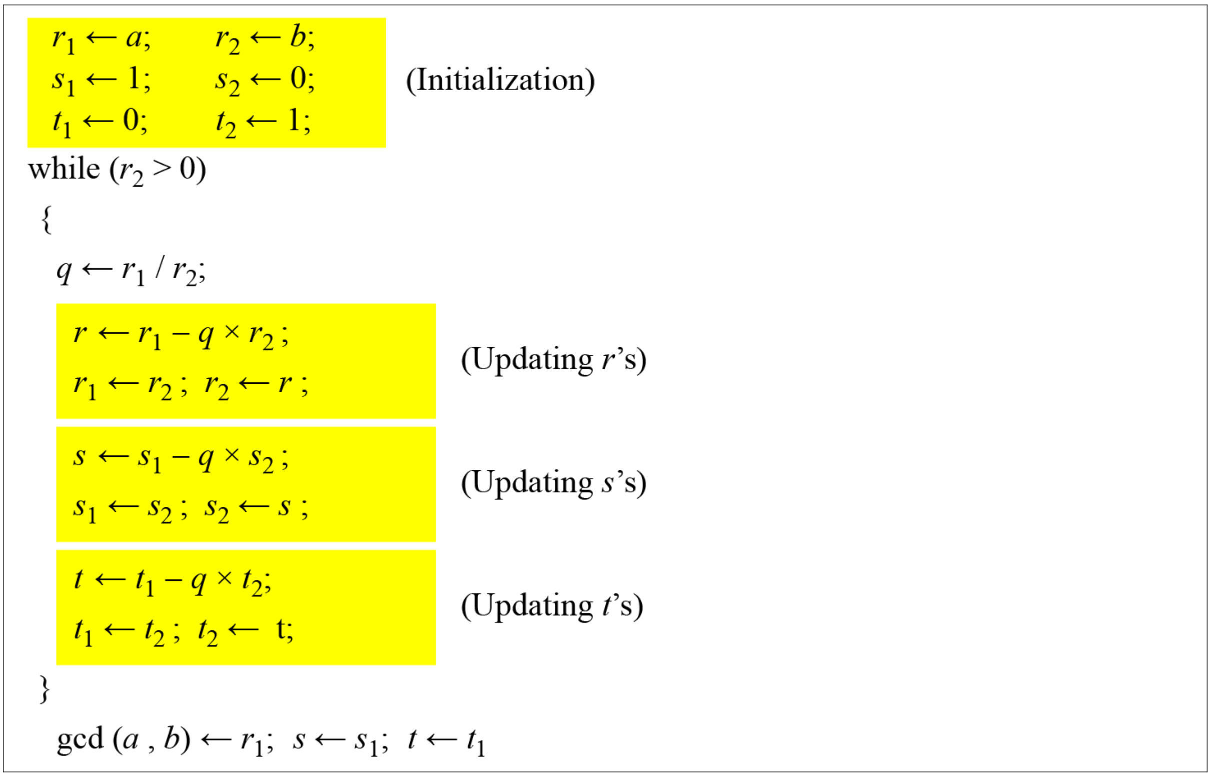
Given two integers a and b, we often need to find other two integers, s and t, such that -

$$s \times a + t \times b = \gcd\ (a, b)$$

The extended Euclidean algorithm can calculate the gcd (a, b) and at the same time calculate the value of s and t.



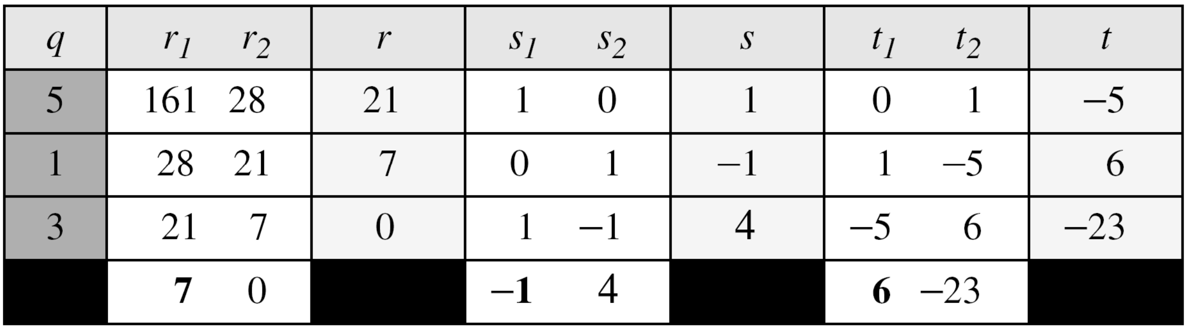
a. Process



```
$r_1 \leftarrow a$;        $r_2 \leftarrow b$;
$s_1 \leftarrow 1$;        $s_2 \leftarrow 0$;        (Initialization)
$t_1 \leftarrow 0$;        $t_2 \leftarrow 1$;
while ($r_2 > 0$)
{
    $q \leftarrow r_1 / r_2$;
    $r \leftarrow r_1 - q \times r_2$;
    $r_1 \leftarrow r_2$; $r_2 \leftarrow r$;        (Updating $r$'s)
    $s \leftarrow s_1 - q \times s_2$;
    $s_1 \leftarrow s_2$; $s_2 \leftarrow s$;        (Updating $s$'s)
    $t \leftarrow t_1 - q \times t_2$;
    $t_1 \leftarrow t_2$; $t_2 \leftarrow$ t;        (Updating $t$'s)
}
gcd ($a$ , $b$) $\leftarrow r_1$;  $s \leftarrow s_1$;  $t \leftarrow t_1$
```

b. Algorithm

Given a = 161 and b = 28, find gcd (a, b) and the values of s and t.



q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	−5
1	28	21	7	0	1	−1	1	−5	6
3	21	7	0	1	−1	4	−5	6	−23
	7	0		**−1**	4		**6**	−23	

We get gcd (161, 28) = 7, s = −1 and t = 6.

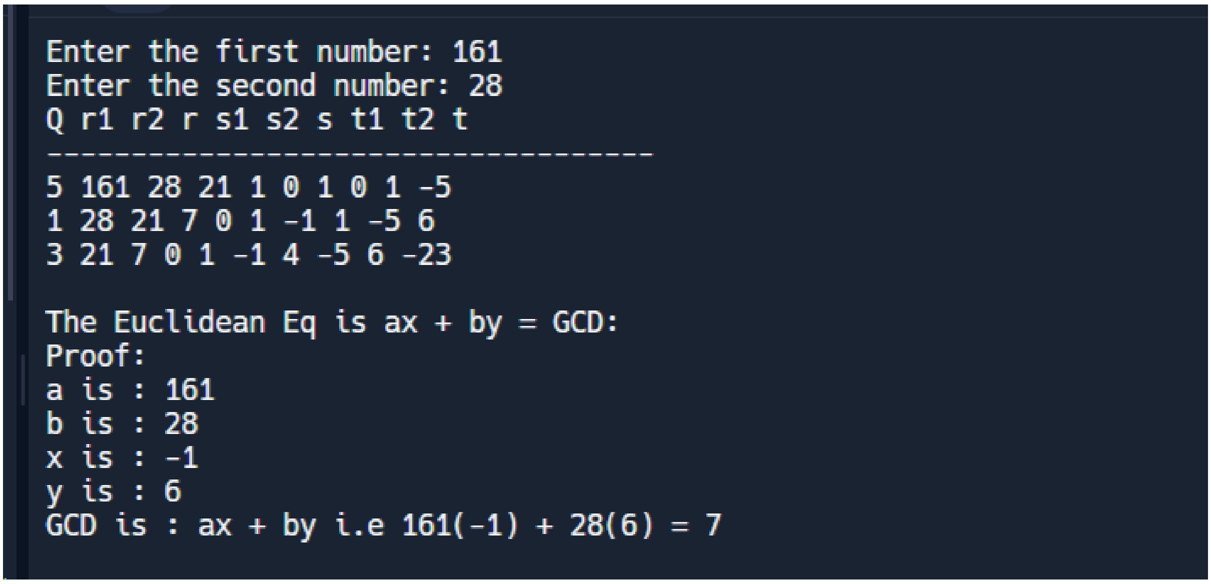
The extended Euclidean algorithm finds the multiplicative inverses of b in Z_n when n and b are given and gcd (n, b) = 1.

The multiplicative inverse of b is the value of t after being mapped to Z_n.

Code:

```
#include <iostream>
using namespace std;

int main() {
  int a, b;
  cout << "Enter the first number: ";
  cin >> a;
  cout << "Enter the second number: ";
  cin >> b;

  int r1 = max(a, b);
  int r2 = min(a, b);
  a = r1;
  b = r2;

  int s1 = 1;
  int s2 = 0;
  int t1 = 0;
  int t2 = 1;

  cout << "Q r1 r2 r s1 s2 s t1 t2 t" << endl;
  cout << "------------------------------------" << endl;

  while (r2 > 0) {
    int q = r1 / r2;
    int rem = r1 % r2;
    int s = s1 - (q * s2);
    int t = t1 - (q * t2);

    cout << q << " " << r1 << " " << r2 << " " << rem << " " << s1 << " " << s2
         << " " << s << " " << t1 << " " << t2 << " " << t << endl;

    r1 = r2;
    r2 = rem;
    s1 = s2;
    s2 = s;
    t1 = t2;
    t2 = t;
  }

  cout << endl;
  cout << "The Euclidean Eq is ax + by = GCD: " << endl;
  cout << "Proof: " << endl;
  cout << "a is : " << a << endl;
  cout << "b is : " << b << endl;
  cout << "x is : " << s1 << endl;
  cout << "y is : " << t1 << endl;
  cout << "GCD is : ax + by i.e " << a << "(" << s1 << ") + " << b << "(" << t1
      << ") = " << (a * s1 + b * t1) << endl;

  return 0;
}
```

Output:



```
Enter the first number: 161
Enter the second number: 28
Q r1 r2 r s1 s2 s t1 t2 t
------------------------------------
5 161 28 21 1 0 1 0 1 -5
1 28 21 7 0 1 -1 1 -5 6
3 21 7 0 1 -1 4 -5 6 -23

The Euclidean Eq is ax + by = GCD:
Proof:
a is : 161
b is : 28
x is : -1
y is : 6
GCD is : ax + by i.e 161(-1) + 28(6) = 7
```