Laboratory 2: Extended Euclidean Algorithm

```
%{
    Name: NE
    Surname: Ramashia
    StudentNo: 1490804
    Date: 04-May-2022
    Lab: Extended Euclidean Algorithm
%}
```

```
Part 1
 [g,a,b] = Extended_Euclidean_Int (240, 46)
 g = int32
    2
 a = int32
    -9
 b = int32
     47
 [g2,a2,b2] = gcd(240,46)
 g2 = 2
 a2 = -9
 b2 = 47
 % Testing the function with 100 sets of inputs
 random_nums = randi([0 1000],100,2);
 isWorking = true;
 for i=1:100
  v = random_nums(i,1);
  u = random_nums(i,2);
  [g,a,b] = Extended_Euclidean_Int (v, u);
  [g2,a2,b2] = gcd(v,u);
  if(g~=g2 || a~=a2 || b~=b2)
      isWorking = false;
  end
 end
```

if isWorking

else

end

fprintf("It is Working");

fprintf("It is NOT woring");

It is Working

The above tests shows that the implemented function generates the correct GCD as well the correct Bézout's identity coefficients.

Part 2

2.
$$Gf(2^{4})$$
, $p=2$, $m=4$

$$p(x) = x^{4} + x + 1$$

$$p(\alpha) = \alpha^{4} + \alpha + 1 = 0$$

$$\alpha^{4} = \alpha + 1$$

*
$$\alpha^6 + \alpha^{'0} = \alpha^2 \alpha^4 + \alpha^4 \alpha^4 \alpha^2$$

= $\alpha^2 (\alpha + 1) + (\alpha + 1)(\alpha + 1) \alpha^4$
= $\alpha^3 + \alpha^2 + \alpha^4 + \alpha^2$
= $\alpha^4 + \alpha^3$
= $\alpha^3 + \alpha + 1$

$$\alpha^{6} = \alpha^{2}\alpha^{4} = \alpha^{2}(\alpha+1) = \alpha^{3} + \alpha^{2}$$

$$\alpha^{6} = \alpha^{4}\alpha^{6} = (\alpha+1)(\alpha^{3}+\alpha^{2})$$

$$= \alpha^{4}+\alpha^{3}+\alpha^{3}+\alpha^{2}$$

$$= \alpha^{2}+\alpha+1$$

$$\Rightarrow \alpha^{6} \alpha^{10} = [\alpha^{3} + \alpha^{2}][\alpha^{2} + \alpha + 1]$$

$$= \alpha^{5} + \alpha^{4} + \alpha^{3} + \alpha^{4} + \alpha^{3} + \alpha^{2}$$

$$= \alpha^{5} + \alpha^{2} = \alpha(1 + \alpha) + \alpha^{2}$$

$$= \alpha$$

*
$$\frac{\alpha^6}{\alpha^{10}} = \frac{\alpha^6}{\alpha^6 \alpha^4} = \frac{1}{\alpha^4} = \frac{1}{1+\alpha}$$

```
p = 2;
m = 4;
field = gftuple((-1:p^m-2)',m,p)
field = 16 \times 4
    0
         0
              0
                    0
         0
              0
                    0
    1
    0
         1
              0
                    0
              1
    0
         0
                    0
            0
    0
         0
                    1
    1
         1
              0
                    0
            1
    0
         1
                    0
                 1
1
    0
         0
            1
0
1
              1
    1
         1
    1
         0
              1
                    0
sum = gfadd(6,10,field)
sum = 7
prod = gfmul(6,10,field)
prod = 1
quot = gfdiv(6,10,field)
quot = 11
```

Part 3

```
3) P(x) = 1 + \alpha^{3}x + \alpha^{5}x^{8}

Q(x) = \alpha^{6}x^{3} + \alpha^{2}x^{5}

let \alpha = \alpha^{3}x, b = \alpha^{5}x^{8}, c = \alpha^{6}x^{3}, d = \alpha^{2}x^{5}

\Rightarrow P \times Q = (1 + q + b)(c + d)

= c + d + \alpha c + \alpha d + b c + b d

= \alpha^{6}x^{3} + \alpha^{2}x^{6} + \alpha^{3}x^{1}\alpha^{6}x^{3} + \alpha^{3}x\alpha^{2}x^{5}

+ \alpha^{5}x^{8}\alpha^{6}x^{3} + \alpha^{5}x^{4}\alpha^{5}x^{5} + \alpha^{5}x^{6}

= \alpha^{6}x^{3} + \alpha^{7}x^{4} + \alpha^{7}x^{2} + \alpha^{5}x^{6}

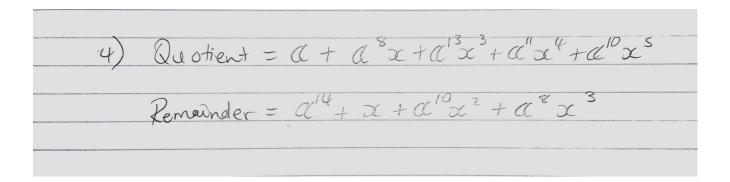
+ \alpha^{11}x^{11} + \alpha^{7}x^{13}
```

```
P3 = [1 3 -Inf -Inf -Inf -Inf -Inf 5];
Q3 = [-Inf -Inf -Inf 6 -Inf 2];
prod3 = gfconv(P3, Q3, field)

prod3 = 1x14
-Inf -Inf -Inf 7 9 3 5 -Inf -Inf -Inf -Inf 11 -Inf ...
```

Part 4

```
quotient4 = 1x10
    11    6    10    13     3    10     1     1     13     0
remainder4 = 1x4
    13    7     3     13
```



Part 5

```
p=2;m=4;
field = gftuple((-1:p^m-2)',m,p);
```

A way to test the function "Extended_Euclidean_GF" was still to be successfully devised.