Galois Groups

May 4, 2022

Contents

Preface

This is my CATAM project, 16.1 for part II. The code for each question can be found in section 4.

1 Question 1

See the table below for some outputs for my programs which calculates the quotient q remainder r when dividing polynomials f by g over $\mathbb{F}_p$.

f	g	q	r	p
$x^5 - 11x^3 + 22x - 11$	$x^4 - x^3 - 4x + 16$	$x + 1$	$4x^3 + 4x^2 + 3x + 1$	7
$x^5 - 11x^3 + 22x - 12$	$x^3 + x^2 - 2x - 1$	$x^2 + 4x + 2$	$2x^2 + 1$	5
$x^4 - x^3 - 4x + 16$	$x + 1$	$x^3 + 11x^2 + 2x + 7$	9	13
$x^3 + 3x + 1$	$x^2 + 3$	x	1	2

Table 1: Various outputs of polynomial division program

See the table below for some outputs for my programs which calculates the GCD of two polynomials f, g over $\mathbb{F}_p$.

f	g	GCD	p
$x^2 + x + 1$	$x + 1$	1	3
$x^2 + 2x + 1$	$x^3 + 3x^2 + 3x + 1$	$x^2 + 2x + 1$	17
$x^6 + 5x^3 + 6x + 1$	$x^5 + 3x^3 + 12x + 7$	$3x + 4$	11
$x^6 + 1$	$x^3 + 1$	2	23

Table 2: Various outputs of polynomial GCD program

A way to efficiently calculate the power, say n, of a polynomial f modulo a polynomial g using my programs written in this question is as follows:

1. Firstly write n in base 2, say $n = b_n \dots b_0 = \sum_{i=0}^{n} b_i 2^i$.
2. Iteratively calculate $f^{2^i} \pmod{g}$ via $f^{2^i} = (f^{2^{i-1}})^2$. Then use the polynomial division algorithm to reduce it modulo g.
3. Iteratively calculate $\prod_{i=0}^{k} f^{b_i 2^i} \pmod{g}$ by consecutively multiplying it by $f^{2^{k+1}}$ if $b_{k+1} = 1$ and 1 otherwise. At each stage use the polynomial division program to reduce the expression modulo g.

2 Question 3

Find below the tables of the decomposition group of each polynomial for each prime between 2 and 97.

Polynomial	Prime												
	2	3	5	7	11	13	17	19	23	29	31	37	41
$x^2 + x + 41$	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_1
$x^3 + 2x + 1$	C_2	C_3	C_3	C_3	C_2	C_2	C_1	C_3	C_2	C_3	C_2	C_2	C_3
$x^3 + x^2 - 2x - 1$	C_3	C_3	C_3	-	C_3	C_1	C_3	C_3	C_3	C_1	C_3	C_3	C_1
$x^4 - 2x^2 + 4$	-	-	C_2	C_2	C_2	C_2	C_2	C_1	C_2	C_2	C_2	C_2	C_2
$x^4 - x^3 - 4x + 16$	-	-	C_4	C_4	-	C_2	C_2	C_4	C_2	C_2	C_2	C_2	C_2
$x^4 - 2x^3 + 5x + 5$	C_4	-	-	C_4	C_4	C_3	C_3	C_3	C_4	C_3	C_3	C_4	C_3
$x^4 + 7x^2 + 6x + 7$	-	-	C_2	C_1	C_2	-	C_2	C_1	C_2	C_2	C_1	C_1	C_2
$x^4 + 3x^3 - 6x^2 - 9x + 7$	-	C_2	-	C_2	C_2	C_2	C_2	C_2	C_2	C_2	C_1	C_2	-
$x^5 + 36$	-	-	-	C_4	C_5	C_4	C_4	C_2	C_4	C_2	C_1	C_4	C_5
$x^5 - 5x + 3$	C_6	C_2	-	-	C_3	C_2	C_2	C_2	C_6	C_3	C_2	C_6	C_2
$x^5 + x^3 - 3x^2 + 3$	-	-	C_5	C_3	C_3	C_5	C_5	C_5	C_5	C_2	C_2	C_5	-
$x^5 - 11x^3 + 22x - 11$	C_5	C_5	C_5	C_5	-	C_5	C_5	C_5	C_1	C_5	C_5	C_5	C_5
$x^6 + x + 1$	C_6	C_6	C_3	C_5	C_6	C_6	C_6	C_4	C_3	C_6	C_6	C_6	C_4
$x^7 - 2x^6 + 2x + 2$	-	-	C_7	C_7	-	C_6	C_4	C_4	C_7	C_7	C_7	C_7	C_5
$x^7 + x^4 - 2x^2 + 8x + 4$	-	-	C_6	C_6	C_2	C_2	C_2	C_3	C_2	C_6	C_2	C_2	C_6
$x^7 + x^5 - 4x^4 - x^3 + 5x + 1$	C_7	-	C_2	C_7	C_2	C_2	C_7	C_7	C_2	C_7	C_7	C_7	C_7

Table 3: Decomposition groups of various polynomials for primes between 2 and 41

Polynomial	Prime											
	43	47	53	59	61	67	71	73	79	83	89	97
x^2+x+41	C_1	C_1	C_1	C_2	C_1	C_2	C_1	C_2	C_2	C_1	C_2	C_1
x^3+2x+1	C_2	C_2	C_3	-	C_2	C_2	C_1	C_2	C_3	C_2	C_2	C_2
x^3+x^2-2x-1	C_1	C_3	C_3	C_3	C_3	C_3	C_1	C_3	C_3	C_1	C_3	C_1
x^4-2x^2+4	C_1	C_2	C_2	C_2	C_2	C_1	C_2	C_1	C_2	C_2	C_2	C_1
$x^4-x^3-4x+16$	C_4	C_2	C_4	C_2	C_2	C_2	C_2	C_2	C_4	C_2	C_4	C_1
x^4-2x^3+5x+5	C_2	C_3	C_2	C_3	C_4	C_3	C_3	C_3	-	C_4	C_4	C_2
x^4+7x^2+6x+7	C_1	C_2	C_2	C_2	C_1	C_1	C_2	C_1	C_1	C_2	C_2	C_1
$x^4+3x^3-6x^2-9x+7$	C_2	C_2	C_2	C_2	C_2	C_2	C_1	C_2	C_1	C_2	C_1	C_2
x^5+36	C_4	C_4	C_4	C_2	C_5	C_4	C_5	C_4	C_2	C_4	C_2	C_4
x^5-5x+3	C_2	C_2	C_2	C_2	C_6	C_3	C_2	C_3	C_2	C_2	C_4	C_5
$x^5+x^3-3x^2+3$	C_3	C_3	C_5	C_3	C_5	C_3	C_2	C_2	C_3	C_3	C_3	C_3
$x^5-11x^3+22x-11$	-	C_5	C_5	C_5	C_5	C_1	C_5	C_5	C_5	C_5	C_1	C_5
x^6+x+1	C_6	C_6	C_4	C_4	C_6	C_5	C_4	C_4	C_6	C_5	C_4	C_3
x^7-2x^6+2x+2	C_5	C_7	C_4	C_7	C_4	C_5	C_7	C_3	C_4	C_7	C_5	C_4
$x^7+x^4-2x^2+8x+4$	C_2	C_2	C_6	-	C_2	C_2	C_2	C_2	C_6	C_2	C_2	C_2
$x^7+x^5-4x^4-x^3+5x+1$	C_7	C_7	C_7	C_2	C_2	C_2	C_2	C_7	C_2	C_7	C_7	C_2

Table 4: Decomposition groups of various polynomials for primes between 43 and 97

3 Question 4

3.1 x^2+x+41

Modulo 2 this polynomial has Galois Group C_2. Hence over $\mathbb{Q}$ it also has Galois group C_2 since this is the biggest group a degree 2 polynomial can have.

3.2 x^3+2x+1

We can see over $\mathbb{Q}$ this polynomial has a Galois group which contains a 2 and 3. Hence it contains a $2, \deg f - 1 = 2$ and $\deg f$ cycle and so has S_3 as it's Galois group.

3.3 x^3+x^2-2x-1

We can see over $\mathbb{Q}$ that this polynomial must contain a 3 cycle It maximally can have size 6 so either the group is D_6 or C_6 or C_3.

3.4 x^4-2x^2+4

We know that it's Galois group over $\mathbb{Q}$ contains a double transposition Hence it's Galois group over $\mathbb{Q}$ must be a subgroup of S_4 which contains C_2 as a subgroup. Further we know this inclusion is strict since if it's $\mathrm{Gal}(f)$ over $\mathbb{Q}$ was C_2 then it would be a reducible polynomial as all degree 2 extensions are quadratic. Then since x^4-2x^2+4 is irreducible, the degree of the extension can't be two.

3.5 $x^4-x^3-4x+16$

We know that it's Galois group over $\mathbb{Q}$ must be a subgroup of S_4 which contains a 4-cycle, a double transposition and single transposition.

3.6 x^4-2x^3+5x+5

We can see that $\mathrm{Gal}(f)$ over $\mathbb{Q}$ must contain C_3, C_4 as a subgroup, further we know it must contain a 2-cycle since it's Galois group over $\mathbb{F}_{43}$ is generated by such a cycle. Hence $\mathrm{Gal}(f)$ over $\mathbb{Q}$ contains a $\deg f, \deg f - 1$ cycle and a transposition and so must S_4.

3.7 $x^4 + 7x^2 + 6x + 7$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^4 + 7x^2 + 6x + 7 = (x^2 - x + 7)(x^2 + x + 1).$$

Further note over $\mathbb{F}_{17}$ it's Galois group is generated by a double transposition and so it's Galois group over $\mathbb{Q}$ must contain a double transposition. But it's the orbits of the action of $\text{Gal}(f)$ on it's roots correspond to it's irreducible factors so we must in fact have that it's Galois group is $C_2 \times C_2$.

3.8 $x^4 + 3x^3 - 6x^2 - 9x + 7$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^4 + 3x^3 - 6x^2 - 9x + 7 = (x^2 + x - 1)(x^2 + 2x - 7).$$

Similarly to the previous polynomial we have the Galois group of this polynomial over $\mathbb{Q}$ must be $C_2 \times C_2$.

3.9 $x^5 + 36$

We can see that over $\mathbb{Q}$, the Galois group of this polynomial contains a double transposition, a 4 cycle and a 5 cycle. Hence the Galois group of the polynomial must be a subgroup of S_5 which contains C_5, C_4 as subgroup.

3.10 $x^5 - 5x + 3$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^5 - 5x + 3 = (x^2 + x - 1)(x^3 - x^2 + 2x - 3).$$

Then it's Galois group over $\mathbb{Q}$ contains C_6 as a subgroup. Hence $Gal(f)$ over $\mathbb{Q}$ is either $C_6, C_2 \times D_{12}$.

3.11 $x^5 + x^3 - 3x^2 + 3$

We can see that over $\mathbb{Q}$, the Galois group of this polynomial contains C_2, C_3, C_5 as subgroups. Hence the Galois group of this polynomial over $\mathbb{Q}$ is a subgroup of S_5 containing a $3, 5$ cycle and a double transposition. Since the finite fields over which this polynomial has C_2 as it's Galois group is generated by a double transposition we can't be sure if $\text{Gal}(f)$ is S_5, which would be the case if $\text{Gal}(f)$ over $\mathbb{Q}$ contains a single transposition.

3.12 $x^5 - 11x^3 + 22x - 11$

We can see that over $\mathbb{Q}$ $\text{Gal}(f)$ contains a C_5 a subgroup. Hence $\text{Gal}(f)$ over $\mathbb{Q}$ is a subgroup of S_5 containing C_5

3.13 $x^6 + x + 1$

We can see that $\text{Gal}(f)$ over $\mathbb{Q}$ is a subgroup of S_6 containing C_4, C_5, C_6 as subgroups.

3.14 $x^7 - 2x^6 + 2x + 2$

We can see that $\text{Gal}(f)$ over $\mathbb{Q}$ is a subgroup of S_6 containing C_4, C_5, C_6, C_7 as subgroups.

3.15 $x^7 + x^4 - 2x^2 + 8x + 4$

First note this polynomial is reducible, indeed decomposing it into irreducible factors gives

$$x^7 + x^4 - 2x^2 + 8x + 4 = (x^3 + 2x + 1)(x^4 - 2x^2 + 4).$$

Then it's Galois group over $\mathbb{Q}$ contains C_6 as a subgroup. Note as previously discussed $\text{Gal}_{\mathbb{Q}}(x^3 + 2x + 1) = D_6$ and $C_2 < \text{Gal}_{\mathbb{Q}}(x^4 - 2x^2 + 4)$. Then certainly $\text{Gal}(f)$ over $\mathbb{Q}$ must contain a copy of D_6 and C_6.

3.16 $x^7 + x^5 - 4x^4 - x^3 + 5x + 1$

We can see that $\text{Gal}(f)$ over $\mathbb{Q}$ must contain C_2 and C_7 as a subgroup.

3.17 Frequency of Cycle Types

I conjecture that the frequency of cycle types that occurs in the Galois group of a polynomial over $\mathbb{F}_p$ for p primes between 1 and N is roughly fixed as N gets large. Below I've included two tables, both giving the frequency of cycle types for polynomials over F_p between 1 and N. The first table gives this for $N = 100$ and the second for $N = 31$. The data given supports my hypothesis with exception of $f(x) = x^2 + x + 41$. The ratio of a 1 cycle to a 2 cycle occurring for $N = 1000$ is $0.43 : 0.57$ which does in fact support my conjecture.

Polynomial	Cycle Type Frequency						
	1	2	3	4	5	6	7
$x^2 + x + 41$	0.32	0.68	0	0	0	0	0
$x^3 + 2x + 1$	0.08	0.58	0.33	0	0	0	0
$x^3 + x^2 - 2x - 1$	0.29	0	0.71	0	0	0	0
$x^4 - 2x^2 + 4$	0.22	0.78	0	0	0	0	0
$x^4 - x^3 - 4x + 16$	0.05	0.64	0	0.32	0	0	0
$x^4 - 2x^3 + 5x + 5$	0	0.14	0.5	0.36	0	0	0
$x^4 + 7x^2 + 6x + 7$	0.45	0.55	0	0	0	0	0
$x^4 + 3x^3 - 6x^2 - 9x + 7$	0.18	0.82	0	0	0	0	0
$x^5 + 36$	0.05	0.23	0	0.55	0.18	0	0
$x^5 - 5x + 3$	0	0.08	0.08	0.29	0.25	0.29	0
$x^5 + x^3 - 3x^2 + 3$	0	0.18	0.45	0	0.36	0	0
$x^5 - 11x^3 + 22x - 11$	0.13	0	0	0	0.87	0	0
$x^6 + x + 1$	0	0	0.12	0.28	0.12	0.48	0
$x^7 - 2x^6 + 2x + 2$	0	0	0.05	0.27	0.18	0.05	0.45
$x^7 + x^4 - 2x^2 + 8x + 4$	0	0.68	0.05	0	0	0.27	0
$x^7 + x^5 - 4x^4 - x^3 + 5x + 1$	0	0.42	0	0	0	0	0.58

Table 5: Frequency of Cycle Types of Galois Groups of Polynomials Over Primes

Polynomial	Cycle Type Frequency						
	1	2	3	4	5	6	7
$x^2 + x + 41$	0	1	0	0	0	0	0
$x^3 + 2x + 1$	0.09	0.45	0.45	0	0	0	0
$x^3 + x^2 - 2x - 1$	0.2	0	0.8	0	0	0	0
$x^4 - 2x^2 + 4$	0.11	0.89	0	0	0	0	0
$x^4 - x^3 - 4x + 16$	0	0.63	0	0.38	0	0	0
$x^4 - 2x^3 + 5x + 5$	0	0	0.56	0.44	0	0	0
$x^4 + 7x^2 + 6x + 7$	0.38	0.63	0	0	0	0	0
$x^4 + 3x^3 - 6x^2 - 9x + 7$	0.11	0.89	0	0	0	0	0
$x^5 + 36$	0.13	0.25	0	0.5	0.13	0	0
$x^5 - 5x + 3$	0	0.1	0.1	0.3	0.1	0.4	0
$x^5 + x^3 - 3x^2 + 3$	0	0.22	0.22	0	0.56	0	0
$x^5 - 11x^3 + 22x - 11$	0.1	0	0	0	0.9	0	0
$x^6 + x + 1$	0	0	0.18	0.09	0.09	0.64	0
$x^7 - 2x^6 + 2x + 2$	0	0	0	0.25	0	0.13	0.63
$x^7 + x^4 - 2x^2 + 8x + 4$	0	0.56	0.11	0	0	0.33	0
$x^7 + x^5 - 4x^4 - x^3 + 5x + 1$	0	0.4	0	0	0	0	0.6

Table 6: Frequency of Cycle Types of Galois Groups of Polynomials Over Primes between 2 and 31

4 Code

4.1 Question 1

```
function [Division] = poldiv(n,d,p)
%Divides polynomial n by polynomial d modulo p
%Note this program makes use of the DocPolynom package and polynomials
```

```
%should be given in the DocPolynom form

n=DocPolynom(mod(double(n),p));
d=DocPolynom(mod(double(d),p));



zero=DocPolynom([0]);
q=zero;
r=n;
vd=double(d);
vq=double(q);
vr= double(r);

dd=size(vd,2)-1;
dr=size(vr,2)-1;
j=1;


while norm(vr) ~= 0 && dd<=dr
    lr=vr(1);
    ld=vd(1);
    a=ld;
        C=0;
        i=1;
        while i <= p-1 && C==0
            if mod(a*i,p)==1
                ldinv=i;
                C=1;
            else
            end
            i=i+1;
        end
        dr;
        dd;
    t=zeros(1,dr-dd+1);
    t(1)=(mod(lr*ldinv,p));
    t=DocPolynom(t);
    t;
    q=plus(q,t);
    r=minus(r,mtimes(t,d));
    q=DocPolynom(mod(double(q),p));
    r=DocPolynom(mod(double(r),p));

    vq=double(q);
    vr= double(r);
    r;
    norm(vr);
    dd;
    dr;
    vr;

    dq=size(vq,2)-1;
    dr=size(vr,2)-1;



end

    q
```

```
    d
    r
    DocPolynom(mod(double(plus(r,mtimes(q,d))),p))








end
```

```
function [outputArg1,outputArg2] = polgcd(a,b,p)
%finds the gcd of polynomials a,b (given in DocPolynom format) over the
%field F_p

a=DocPolynom(mod(double(a),p));
b=DocPolynom(mod(double(b),p));
va=double(a);
vb=double(b);
da=size(va,2);
db=size(vb,2);
%if da>db
   % holding=b;
    %b=a;
    %a=holding;
%end
a=DocPolynom(mod(double(a),p));
b=DocPolynom(mod(double(b),p));
va=double(a);
vb=double(b);
da=size(va,2);
db=size(vb,2);


index=0;
divalg=0;
while norm(vb) ~= 0
    T=b;
    a;
    b;
    % Run divisor algorithm
            n=DocPolynom(mod(double(a),p));
            d=DocPolynom(mod(double(b),p));
            n;
            d;


            zero=DocPolynom([0]);
            q=zero;
            r=n;
            vd=double(d);
            vq=double(q);
            vr= double(r);

            dd=size(vd,2)-1;
            dr=size(vr,2)-1;
            j=1;

```

```

            while norm(vr) ~= 0 && dd<=dr
                lr=vr(1);
                ld=vd(1);
                a=ld;
                    C=0;
                    i=1;
                    while i <= p-1 && C==0
                        if mod(a*i,p)==1
                            ldinv=i;
                            C=1;
                        else
                        end
                        i=i+1;
                    end
                    dr;
                    dd;
                t=zeros(1,dr-dd+1);
                t(1)=(mod(lr*ldinv,p));
                t=DocPolynom(t);
                t;
                q=plus(q,t);
                r=minus(r,mtimes(t,d));
                q=DocPolynom(mod(double(q),p));
                r=DocPolynom(mod(double(r),p));

                vq=double(q);
                vr= double(r);


                dq=size(vq,2)-1;
                dr=size(vr,2)-1;




                q;
                r;
            end

    % Finished running divisor algorithm
    b=DocPolynom(mod(double(r),p));
    a=DocPolynom(mod(double(T),p));
    va=double(a);
    vb=double(b);
    da=size(va,2);
    db=size(vb,2);
end
a




end
```

4.2 Question 2

```
function [outputArg1,outputArg2] = decompcalc4(f)
%Calcualtes decomposition group of f from primes 2 to 97
```

```
%Preparation
        primes=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
             71, 73, 79, 83, 89, 97];
       % primes=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61,
           67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139,
           149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227,
           229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311,
           313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401,
           409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491,
           499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599,
           601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683,
           691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797,
           809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887,
           907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997];


        %primes=ones(1,25);
        %primes=5*primes;
         pindex=1;
                 holdf=f;
                 vholdf = double(holdf);
                 dholdf = size(vholdf,2)-1;
                 G_pf=zeros(size(primes,2), dholdf );
                 Group=zeros(size(primes,2),1);


while pindex <= size(primes,2)
    cou=1;
    %Setup
        p=primes(pindex);
                 vf=mod(double(holdf),p);
                 f=DocPolynom(vf);
                 df=size(vf,2)-1;

    %Check if (f,Df)=1










                 %Step 1: Checking if (f,f')=1

                                 %Calculate f'=Df

                                         vDf=zeros(1,df);
                                     i=1;
                                     while i<= df
                                         vDf(i)= (df-i+1)*vf(i);
                                         i=i+1;
                                     end

                                     Df=DocPolynom(vDf);
                                 %Calculated Df

```

```
                                %Run gcd program on (f,Df)

                                        a=f;
                                        b=Df;


                                                %finds the gcd of polynomials a,b (
                                                   given in DocPolynom format) over
                                                   the
                        %field F_p

                            a=DocPolynom(mod(double(a),p));
                            b=DocPolynom(mod(double(b),p));
                            va=double(a);
                            vb=double(b);
                            da=size(va,2);
                            db=size(vb,2);
                            if da<db
                                holding=a;
                                a=b;
                                b=holding;
                            end
                            a=DocPolynom(mod(double(a),p));
                            b=DocPolynom(mod(double(b),p));
                            va=double(a);
                            vb=double(b);
                            da=size(va,2);
                            db=size(vb,2);


                            index=0;
                            divalg=0;
                            while norm(vb) ~= 0
                                T=b;
                                % Run divisor algorithm
                                        n=DocPolynom(mod(double(a),p));
                                        d=DocPolynom(mod(double(b),p));



                                        zero=DocPolynom([0]);
                                        q=zero;
                                        r=n;
                                        vd=double(d);
                                        vq=double(q);
                                        vr= double(r);

                                        dd=size(vd,2)-1;
                                        dr=size(vr,2)-1;
                                        j=1;


                                        while norm(vr) ~= 0 && dd<=dr
                                            lr=vr(1);
                                            ld=vd(1);
                                            ae=ld;
                                                C=0;
                                                i=1;
                                                while i <= p-1 && C==0
```

```
                                                    if mod(ae*i,p)==1
                                                        ldinv=i;
                                                        C=1;
                                                    else
                                                    end
                                                    i=i+1;
                                                end
                                                dr;
                                                dd;
                                            t=zeros(1,dr-dd+1);
                                            t(1)=(mod(lr*ldinv,p));
                                            t=DocPolynom(t);
                                            t;
                                            q=plus(q,t);
                                            r=minus(r,mtimes(t,d));
                                            q=DocPolynom(mod(double(q),p));
                                            r=DocPolynom(mod(double(r),p));

                                            vq=double(q);
                                            vr= double(r);


                                            dq=size(vq,2)-1;
                                            dr=size(vr,2)-1;


                                            divalg=divalg+1;
                                            q;
                                            r;
                                        end

                                % Finished running divisor algorithm
                                b=DocPolynom(mod(double(r),p));
                                a=DocPolynom(mod(double(T),p));
                                va=double(a);
                                vb=double(b);
                                da=size(va,2);
                                db=size(vb,2);
                            end



                                        gcd=a;

                                %Ran gcd program


                                %Check if gcd(f,Df)=1
                                        vgcd=double(gcd);


    if size(vgcd,2)==1

        %For each rr compute n_rr

                %Calculating f_rr

                                        v=f;
                                        w=DocPolynom([1 0]);
                                        rr=1;
```

```
                    dv=df;



while 2*rr <= dv



    %efficient program to calculate x^p^rr
     bin=dec2bin(p^rr);
    sizebin=size(bin,2);
    powerindex=2;
    if str2num(bin(sizebin)) == 1
        nn=DocPolynom([1 0]);
    else
        nn=DocPolynom([ 1]);
    end

    nn_i=DocPolynom([1 0]);
    while powerindex <= sizebin;
        nn_i=nn_i^2;
        n=nn_i;
        d=v;
            %run divisor algorithm with n=nn_i, d=v

                n=DocPolynom(mod(double(n),p));
                d=DocPolynom(mod(double(d),p));



                zero=DocPolynom([0]);
                q=zero;
                r=n;
                vd=double(d);
                vq=double(q);
                vr= double(r);

                dd=size(vd,2)-1;
                dr=size(vr,2)-1;
                j=1;


                while norm(vr) ~= 0 && dd<=dr
                    lr=vr(1);
                    ld=vd(1);
                    a=ld;
                        C=0;
                        i=1;
                        while i <= p-1 && C==0
                            if mod(a*i,p)==1
                                ldinv=i;
                                C=1;
                            else
                            end
                            i=i+1;
                        end
                        dr;
                        dd;
                    t=zeros(1,dr-dd+1);
                    t(1)=(mod(lr*ldinv,p));
```

```
            t=DocPolynom(t);
            t;
            q=plus(q,t);
            r=minus(r,mtimes(t,d));
            q=DocPolynom(mod(double(q),p));
            r=DocPolynom(mod(double(r),p));

            vq=double(q);
            vr= double(r);
            r;
            norm(vr);
            dd;
            dr;
            vr;

            dq=size(vq,2)-1;
            dr=size(vr,2)-1;



        end




    %ran divisor algorithm and let nn_i=r
    nn_i=r;
if str2num(bin(sizebin-powerindex+1)) ==1
        nn=mtimes(nn,nn_i);
        n=nn;
        d=v;

        %run divisor algorithm with n=nn and
        %d=v

                n=DocPolynom(mod(double(n),p));
                d=DocPolynom(mod(double(d),p));



                zero=DocPolynom([0]);
                q=zero;
                r=n;
                vd=double(d);
                vq=double(q);
                vr= double(r);

                dd=size(vd,2)-1;
                dr=size(vr,2)-1;
                j=1;


                while norm(vr) ~= 0 && dd<=dr
                    lr=vr(1);
                    ld=vd(1);
                    a=ld;
                        C=0;
                        i=1;
                        while i <= p-1 && C==0
                            if mod(a*i,p)==1
```

```
                                                            ldinv=i;
                                                            C=1;
                                                        else
                                                        end
                                                        i=i+1;
                                                    end
                                                    dr;
                                                    dd;
                                                t=zeros(1,dr-dd+1);
                                                t(1)=(mod(lr*ldinv,p));
                                                t=DocPolynom(t);
                                                t;
                                                q=plus(q,t);
                                                r=minus(r,mtimes(t,d));
                                                q=DocPolynom(mod(double(q),p));
                                                r=DocPolynom(mod(double(r),p));

                                                vq=double(q);
                                                vr= double(r);
                                                r;
                                                norm(vr);
                                                dd;
                                                dr;
                                                vr;

                                                dq=size(vq,2)-1;
                                                dr=size(vr,2)-1;



                                            end


                                    %rand divisor algorithm and let nn=r
                                    nn=r;
                            end
                            powerindex=powerindex+1;
                        end
                        nn=DocPolynom(mod(double(nn),p));
                        w=nn;

                        %end of efficient program to calculate w^p
                        n=nn;
                        d=v;

                        %Run divisor algorithm

                            n=DocPolynom(mod(double(n),p));
                            d=DocPolynom(mod(double(d),p));



                            zero=DocPolynom([0]);
                            q=zero;
                            r=n;
                            vd=double(d);
                            vq=double(q);
                            vr= double(r);

                            dd=size(vd,2)-1;
```

```
        dr=size(vr,2)-1;
        j=1;


        while norm(vr) ~= 0 && dd<=dr
            lr=vr(1);
            ld=vd(1);
            a=ld;
                C=0;
                i=1;
                while i <= p-1 && C==0
                    if mod(a*i,p)==1
                        ldinv=i;
                        C=1;
                    else
                    end
                    i=i+1;
                end
                dr;
                dd;
            t=zeros(1,dr-dd+1);
            t(1)=(mod(lr*ldinv,p));
            t=DocPolynom(t);
            t;
            q=plus(q,t);
            r=minus(r,mtimes(t,d));
            q=DocPolynom(mod(double(q),p));
            r=DocPolynom(mod(double(r),p));

            vq=double(q);
            vr= double(r);


            dq=size(vq,2)-1;
            dr=size(vr,2)-1;



        end


    %Once ran divisor algorithm let w=r

    w=r;






    vw=double(w);
    dw=size(vw,2)-1;


    %Run GCD program on w-x and v
        a=minus(w,DocPolynom([1 0]));
        b=v;

            a=DocPolynom(mod(double(a),p));
            b=DocPolynom(mod(double(b),p));
```

```
va=double(a);
vb=double(b);
da=size(va,2);
db=size(vb,2);
%if da>db
   % holding=b;
    %b=a;
    %a=holding;
%end
a=DocPolynom(mod(double(a),p));
b=DocPolynom(mod(double(b),p));
va=double(a);
vb=double(b);
da=size(va,2);
db=size(vb,2);


index=0;
divalg=0;
while norm(vb) ~= 0
    T=b;
    a;
    b;
    % Run divisor algorithm
            n=DocPolynom(mod(double(a),p));
            d=DocPolynom(mod(double(b),p));
            n;
            d;


            zero=DocPolynom([0]);
            q=zero;
            r=n;
            vd=double(d);
            vq=double(q);
            vr= double(r);

            dd=size(vd,2)-1;
            dr=size(vr,2)-1;
            j=1;


            while norm(vr) ~= 0 && dd<=dr
                lr=vr(1);
                ld=vd(1);
                a=ld;
                    C=0;
                    i=1;
                    while i <= p-1 && C==0
                        if mod(a*i,p)==1
                            ldinv=i;
                            C=1;
                        else
                        end
                        i=i+1;
                    end
                    dr;
                    dd;
                t=zeros(1,dr-dd+1);
                t(1)=(mod(lr*ldinv,p));
```

```
                        t=DocPolynom(t);
                        t;
                        q=plus(q,t);
                        r=minus(r,mtimes(t,d));
                        q=DocPolynom(mod(double(q),p));
                        r=DocPolynom(mod(double(r),p));

                        vq=double(q);
                        vr= double(r);


                        dq=size(vq,2)-1;
                        dr=size(vr,2)-1;




                        q;
                        r;
                    end

            % Finished running divisor algorithm
            b=DocPolynom(mod(double(r),p));
            a=DocPolynom(mod(double(T),p));
            va=double(a);
            vb=double(b);
            da=size(va,2);
            db=size(vb,2);
        end

%Once ran GCD program let g=gcd
g=a;
w;
v;
g;
rr;

vg=double(g);
dg=size(vg,2)-1;
if dg ~= 0

    g;
    rr;
    G_pf(pindex,rr)= dg/rr;

    n=v;
    d=g;
    %Run divisor algorithm

        n=DocPolynom(mod(double(n),p));
        d=DocPolynom(mod(double(d),p));



        zero=DocPolynom([0]);
        q=zero;
        r=n;
        vd=double(d);
        vq=double(q);
        vr= double(r);
```

```
            dd=size(vd,2)-1;
            dr=size(vr,2)-1;
            j=1;


            while norm(vr) ~= 0 && dd<=dr
                lr=vr(1);
                ld=vd(1);
                a=ld;
                    C=0;
                    i=1;
                    while i <= p-1 && C==0
                        if mod(a*i,p)==1
                            ldinv=i;
                            C=1;
                        else
                        end
                        i=i+1;
                    end
                    dr;
                    dd;
                t=zeros(1,dr-dd+1);
                t(1)=(mod(lr*ldinv,p));
                t=DocPolynom(t);
                t;
                q=plus(q,t);
                r=minus(r,mtimes(t,d));
                q=DocPolynom(mod(double(q),p));
                r=DocPolynom(mod(double(r),p));

                vq=double(q);
                vr= double(r);


                dq=size(vq,2)-1;
                dr=size(vr,2)-1;



            end

        %Once ran divisor algorithm let v=q
        v=q;
        vv=double(v);
        dv=size(vv,2)-1;

        %cut the rest out


        %end of cutting out

    end
    rr=rr+1;
end

if dv ~= 0

    v;
    dv;
```

```
                        G_pf(pindex,dv)=1;

                    end

















    else
        G_pf(pindex,:)=(-1)*ones(1,dholdf);
    end

    %Calculate size of group


        if G_pf(pindex,1) == -1
                        Group(pindex,1)=-1;
                else
                        i=1;
                        N=zeros(1,df);
                        while i <= df

                                if G_pf(pindex,i)==0

                                        N(i)=1;

                                else
                                        N(i)=i;

                end
            i=i+1;
            end
            N;
            x=lcm(sym(N));
                        Group(pindex,1)=x;

        end

  pindex=pindex+1;
end
G_pf
Group
end
```