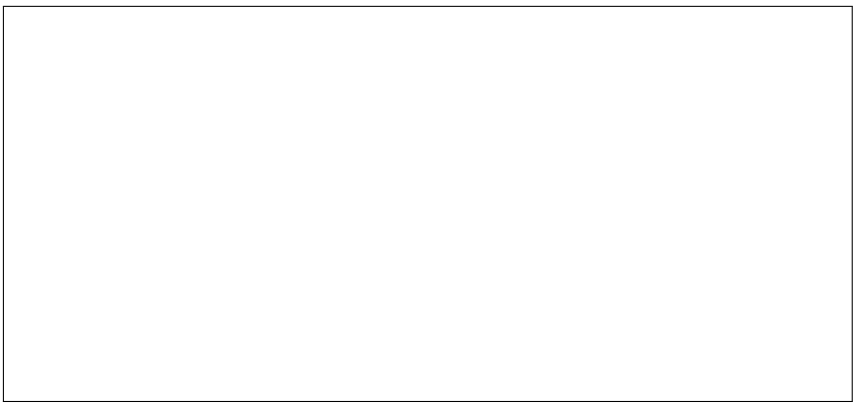
16.1



The Galois Group of a Polynomial

Contents

Introduction

This project is programmed in **Python 3.5**. Consult section A for program documentation, listings and information on the structure of the programming for the project, as appropriate. This report is written in LaTeX 2_ε.

1 Algorithms

Background. Let's first settle some semantics. Let $f \in \mathbb{Z}[X]$ be monic. Via coefficient-wise inclusion, $\mathbb{Z}[X] \hookrightarrow \mathbb{Q}[X]$, and so $f \in \mathbb{Q}[X]$. Let $F(f)$ be a splitting field of $f \in \mathbb{Q}[X]$ (for the sake of concreteness, take this to be $\mathbb{Q}(\mathrm{Root}_f(\mathbb{C})) \leq \mathbb{C}$ with the inclusion map). Then $F(f)/\mathbb{Q}$ is a Galois extension with Galois group $\mathrm{Gal}(F(f)/\mathbb{Q})$.

Let $Z(f) = \mathrm{Root}_f(F(f)) \subseteq F(f)$. Suppose f is *separable*[1] – i.e. $f,\ f' \in \mathbb{Q}[X]$ have no common irreducible factor. Then $|Z(f)| = \deg(f)$.

$\mathrm{Gal}(F(f)/\mathbb{Q})$ acts on $Z(f)$ via $\phi \mapsto (x \mapsto \phi(x))$ (closure since $\forall \phi \in \mathrm{Gal}(F(f)/\mathbb{Q})\ \ \forall x \in Z(f)$ $f(\phi(x)) = \phi(f(x)) = \phi(0) = 0$; other properties are immediate). The permutation representation Ω of this action is injective because $Z(f) \subseteq F(f)^{\ker(\Omega)}$ (the fixed field of $\ker(\Omega)$) and $\mathbb{Q}(Z(f)) = F(f)$, whence $\ker(\Omega) = \{\iota\}$. Hence, via ρ,

$$\mathrm{Gal}(F(f)/\mathbb{Q}) \cong G(f) := \{Z(f) \to Z(f),\ x \mapsto \phi(x) :\ \phi \in \mathrm{Gal}(F(f)/\mathbb{Q})\} \leq \mathrm{Sym}(Z(f)) \cong S_n$$

$G(f)$ is the *Galois group* of f, and inherits cycle types from $\mathrm{Sym}(Z(f))$. We aim to classify it up to isomorphism. Although the distribution of cycle types is not itself a group property, it will prove useful in the computation of the group and is the basis of the project algorithm.

Q1. Let $p \in \mathbb{P}$ and consider the ring $R = \mathbb{Z}_p[X]$. The **division algorithm** is implemented as the function `div`(a, b, p), which returns the quotient and remainder of dividing $a \in R$ by $b \in R\backslash\{0\}$. For example, in $\mathbb{Z}_5[X]$, $X^3 = (3X^2 + X + 2)(2X + 1) + 3$.

```
>>> div([0,0,0,1],[1,2],5)          [[2, 1, 3], [3]]
```

Via **Euclid's algorithm**, this can be applied to compute the highest common factor of $(a, b) \in R^2\backslash\{(0, 0)\}$, whose monic associate is returned by the function `hcf`(a, b, p). For example, noting that all linear polynomials over a field are irreducible,

```
>>> hcf(mp(mp([2,2],[3,3],5),[4,1],5), mp([1,1],[2,1],5),5)
[1, 1]
```

Another application of the division algorithm we require is for a fast **modular exponentiation algorithm**. Implemented as `exp`(a, b, n, p), which computes the remainder of dividing a^n ($a \in R$, $n \in \mathbb{N}_0$) by $b \in R\backslash\{0\}$, the algorithm decomposes n into binary, generates a truncated sequence of a^{2^k} by iterated squaring and reduction modulo b (with the division algorithm) and multiplies its terms together sequentially according to n, reducing modulo b between each step. E.g., in $\mathbb{Z}_5[X]$, $(X + 1)^3 = X^3 + 3X^2 + 3X + 1 \equiv_{X^2} 3X + 1$.

```
>>> exp([1,1],[0,0,1],3,5)          [1, 3]
```

Q2. The decomposition algorithm is implemented as `decomp`(f, p), which computes the cycle type of some generator of the decomposition group of $f \in \mathbb{Z}[X]$ modulo $p \in \mathbb{P}$, viewed as a permutation group in the way outlined in *Background*.[2] The decomposition group itself is thus isomorphic to C_n, where $n \in \mathbb{N}$ is the order of the generator – i.e. the lowest common multiple of its cycle type (as an integer partition).

The notation we'll use to represent cycle types is $[a_i]_{i=1}^k$, where $a_i \in \mathbb{N}_0$ is the number of i-cycles present, and $k \in \mathbb{N}$ is at least the longest cycle's length. Trailing zeros may be omitted.

Q3. Here follows the output of `decomp`(f, p), for f an example polynomial and $p \in \mathbb{P} \cap [1, 97]$.

[1] Warning: a slightly unconventional definition.

[2] So, irrespective of choice of splitting field over $\mathbb{Z}_p[X]$. Interpreting the program's output this way requires one to envisage (but not construct!) a fixed splitting field. However, we only care that the output is the cycle type of some element of $G(f)$.

```
X^2 + X + 41
 2: [0, 1]
 3: [0, 1]
 5: [0, 1]
 7: [0, 1]
11: [0, 1]
13: [0, 1]
17: [0, 1]
19: [0, 1]
23: [0, 1]
29: [0, 1]
31: [0, 1]
37: [0, 1]
41: [2, 0]
43: [2, 0]
47: [2, 0]
53: [2, 0]
59: [0, 1]
61: [2, 0]
67: [0, 1]
71: [2, 0]
73: [0, 1]
79: [0, 1]
83: [2, 0]
89: [0, 1]
97: [2, 0]
```

```
X^3 + X^2 - 2X - 1
 2: [0, 0, 1]
 3: [0, 0, 1]
 5: [0, 0, 1]
 7: Not separable.
11: [0, 0, 1]
13: [3, 0, 0]
17: [0, 0, 1]
19: [0, 0, 1]
23: [0, 0, 1]
29: [3, 0, 0]
31: [0, 0, 1]
37: [0, 0, 1]
41: [3, 0, 0]
43: [3, 0, 0]
47: [0, 0, 1]
53: [0, 0, 1]
59: [0, 0, 1]
61: [0, 0, 1]
67: [0, 0, 1]
71: [3, 0, 0]
73: [0, 0, 1]
79: [0, 0, 1]
83: [3, 0, 0]
89: [0, 0, 1]
97: [3, 0, 0]
```

```
X^3 + 2X + 1
 2: [1, 1, 0]
 3: [0, 0, 1]
 5: [0, 0, 1]
 7: [0, 0, 1]
11: [1, 1, 0]
13: [1, 1, 0]
17: [3, 0, 0]
19: [0, 0, 1]
23: [1, 1, 0]
29: [0, 0, 1]
31: [1, 1, 0]
37: [1, 1, 0]
41: [0, 0, 1]
43: [1, 1, 0]
47: [1, 1, 0]
53: [0, 0, 1]
59: Not separable.
61: [1, 1, 0]
67: [1, 1, 0]
71: [3, 0, 0]
73: [1, 1, 0]
79: [0, 0, 1]
83: [1, 1, 0]
89: [1, 1, 0]
97: [1, 1, 0]
```

```
X^4 - 2X^2 + 4
 2: Not separable.
 3: Not separable.
 5: [0, 2, 0, 0]
 7: [0, 2, 0, 0]
11: [0, 2, 0, 0]
13: [0, 2, 0, 0]
17: [0, 2, 0, 0]
19: [4, 0, 0, 0]
23: [0, 2, 0, 0]
29: [0, 2, 0, 0]
31: [0, 2, 0, 0]
37: [0, 2, 0, 0]
41: [0, 2, 0, 0]
43: [4, 0, 0, 0]
47: [0, 2, 0, 0]
53: [0, 2, 0, 0]
59: [0, 2, 0, 0]
61: [0, 2, 0, 0]
67: [4, 0, 0, 0]
71: [0, 2, 0, 0]
73: [4, 0, 0, 0]
79: [0, 2, 0, 0]
83: [0, 2, 0, 0]
89: [0, 2, 0, 0]
97: [4, 0, 0, 0]
```

```
X^4 - X^3 - 4X + 16
 2: Not separable.
 3: Not separable.
 5: [0, 0, 0, 1]
 7: [0, 0, 0, 1]
11: Not separable.
13: [0, 2, 0, 0]
17: [2, 1, 0, 0]
19: [0, 0, 0, 1]
23: [0, 2, 0, 0]
29: [2, 1, 0, 0]
31: [2, 1, 0, 0]
37: [0, 2, 0, 0]
41: [2, 1, 0, 0]
43: [0, 0, 0, 1]
47: [0, 2, 0, 0]
53: [0, 0, 0, 1]
59: [0, 2, 0, 0]
61: [0, 2, 0, 0]
67: [2, 1, 0, 0]
71: [0, 2, 0, 0]
73: [0, 2, 0, 0]
79: [0, 0, 0, 1]
83: [0, 2, 0, 0]
89: [0, 0, 0, 1]
97: [4, 0, 0, 0]
```

```
X^4 + 7X^2 + 6X + 7
 2: Not separable.
 3: Not separable.
 5: [0, 2, 0, 0]
 7: [4, 0, 0, 0]
11: [0, 2, 0, 0]
13: Not separable.
17: [0, 2, 0, 0]
19: [4, 0, 0, 0]
23: [0, 2, 0, 0]
29: [0, 2, 0, 0]
31: [4, 0, 0, 0]
37: [4, 0, 0, 0]
41: [0, 2, 0, 0]
43: [4, 0, 0, 0]
47: [0, 2, 0, 0]
53: [0, 2, 0, 0]
59: [0, 2, 0, 0]
61: [4, 0, 0, 0]
67: [4, 0, 0, 0]
71: [0, 2, 0, 0]
73: [4, 0, 0, 0]
79: [4, 0, 0, 0]
83: [0, 2, 0, 0]
89: [0, 2, 0, 0]
97: [4, 0, 0, 0]
```

```
X^4 - 2X^3 + 5X + 5
 2: [0, 0, 0, 1]
 3: Not separable.
 5: Not separable.
 7: [0, 0, 0, 1]
11: [0, 0, 0, 1]
13: [1, 0, 1, 0]
17: [1, 0, 1, 0]
19: [1, 0, 1, 0]
23: [0, 0, 0, 1]
29: [1, 0, 1, 0]
31: [1, 0, 1, 0]
37: [0, 0, 0, 1]
41: [1, 0, 1, 0]
43: [2, 1, 0, 0]
47: [1, 0, 1, 0]
53: [0, 2, 0, 0]
59: [1, 0, 1, 0]
61: [0, 0, 0, 1]
67: [1, 0, 1, 0]
71: [1, 0, 1, 0]
73: [1, 0, 1, 0]
79: Not separable.
83: [0, 0, 0, 1]
89: [0, 0, 0, 1]
97: [0, 2, 0, 0]
```

```
X^4 + 3X^3 - 6X^2 - 9X + 7
 2: Not separable.
 3: [0, 2, 0, 0]
 5: Not separable.
 7: [2, 1, 0, 0]
11: [2, 1, 0, 0]
13: [0, 2, 0, 0]
17: [2, 1, 0, 0]
19: [2, 1, 0, 0]
23: [2, 1, 0, 0]
29: [2, 1, 0, 0]
31: [4, 0, 0, 0]
37: [0, 2, 0, 0]
41: Not separable.
43: [0, 2, 0, 0]
47: [2, 1, 0, 0]
53: [0, 2, 0, 0]
59: [2, 1, 0, 0]
61: [2, 1, 0, 0]
67: [0, 2, 0, 0]
71: [4, 0, 0, 0]
73: [2, 1, 0, 0]
79: [4, 0, 0, 0]
83: [0, 2, 0, 0]
89: [4, 0, 0, 0]
97: [2, 1, 0, 0]
```

```
X^5 + 36
 2: Not separable.
 3: Not separable.
 5: Not separable.
 7: [1, 0, 0, 1, 0]
11: [0, 0, 0, 0, 1]
13: [1, 0, 0, 1, 0]
17: [1, 0, 0, 1, 0]
19: [1, 2, 0, 0, 0]
23: [1, 0, 0, 1, 0]
29: [1, 2, 0, 0, 0]
31: [5, 0, 0, 0, 0]
37: [1, 0, 0, 1, 0]
41: [0, 0, 0, 0, 1]
43: [1, 0, 0, 1, 0]
47: [1, 0, 0, 1, 0]
53: [1, 0, 0, 1, 0]
59: [1, 2, 0, 0, 0]
61: [0, 0, 0, 0, 1]
67: [1, 0, 0, 1, 0]
71: [0, 0, 0, 0, 1]
73: [1, 0, 0, 1, 0]
79: [1, 2, 0, 0, 0]
83: [1, 0, 0, 1, 0]
89: [1, 2, 0, 0, 0]
97: [1, 0, 0, 1, 0]
```

```
X^5 + X^3 - 3X^2 + 3
 2: Not separable.
 3: Not separable.
 5: [0, 0, 0, 0, 1]
 7: [2, 0, 1, 0, 0]
11: [2, 0, 1, 0, 0]
13: [0, 0, 0, 0, 1]
17: [0, 0, 0, 0, 1]
19: [0, 0, 0, 0, 1]
23: [0, 0, 0, 0, 1]
29: [1, 2, 0, 0, 0]
31: [1, 2, 0, 0, 0]
37: [0, 0, 0, 0, 1]
41: Not separable.
43: [2, 0, 1, 0, 0]
47: [2, 0, 1, 0, 0]
53: [0, 0, 0, 0, 1]
59: [2, 0, 1, 0, 0]
61: [0, 0, 0, 0, 1]
67: [2, 0, 1, 0, 0]
71: [1, 2, 0, 0, 0]
73: [1, 2, 0, 0, 0]
79: [2, 0, 1, 0, 0]
83: [2, 0, 1, 0, 0]
89: [2, 0, 1, 0, 0]
97: [2, 0, 1, 0, 0]
```

```
X^5 - 5X + 3
 2: [0, 1, 1, 0, 0]
 3: [1, 2, 0, 0, 0]
 5: Not separable.
 7: Not separable.
11: [2, 0, 1, 0, 0]
13: [1, 2, 0, 0, 0]
17: [1, 2, 0, 0, 0]
19: [3, 1, 0, 0, 0]
23: [0, 1, 1, 0, 0]
29: [2, 0, 1, 0, 0]
31: [3, 1, 0, 0, 0]
37: [0, 1, 1, 0, 0]
41: [3, 1, 0, 0, 0]
43: [0, 1, 1, 0, 0]
47: [1, 2, 0, 0, 0]
53: [3, 1, 0, 0, 0]
59: [3, 1, 0, 0, 0]
61: [3, 1, 0, 0, 0]
67: [0, 1, 1, 0, 0]
71: [2, 0, 1, 0, 0]
73: [1, 2, 0, 0, 0]
79: [2, 0, 1, 0, 0]
83: [1, 2, 0, 0, 0]
89: [3, 1, 0, 0, 0]
97: [1, 2, 0, 0, 0]
```

```
X^5 - 11X^3 + 22X - 11
 2: [0, 0, 0, 0, 1]
 3: [0, 0, 0, 0, 1]
 5: [0, 0, 0, 0, 1]
 7: [0, 0, 0, 0, 1]
11: Not separable.
13: [0, 0, 0, 0, 1]
17: [0, 0, 0, 0, 1]
19: [0, 0, 0, 0, 1]
23: [5, 0, 0, 0, 0]
29: [0, 0, 0, 0, 1]
31: [0, 0, 0, 0, 1]
37: [0, 0, 0, 0, 1]
41: [0, 0, 0, 0, 1]
43: Not separable.
47: [0, 0, 0, 0, 1]
53: [0, 0, 0, 0, 1]
59: [0, 0, 0, 0, 1]
61: [0, 0, 0, 0, 1]
67: [5, 0, 0, 0, 0]
71: [0, 0, 0, 0, 1]
73: [0, 0, 0, 0, 1]
79: [0, 0, 0, 0, 1]
83: [0, 0, 0, 0, 1]
89: [5, 0, 0, 0, 0]
97: [0, 0, 0, 0, 1]
```

```
X^6 + X + 1
 2: [0, 0, 0, 0, 0, 1]
 3: [1, 1, 1, 0, 0, 0]
 5: [0, 0, 2, 0, 0, 0]
 7: [1, 0, 0, 0, 1, 0]
11: [1, 1, 1, 0, 0, 0]
13: [0, 0, 0, 0, 0, 1]
17: [1, 1, 1, 0, 0, 0]
19: [0, 1, 0, 1, 0, 0]
23: [0, 0, 2, 0, 0, 0]
29: [1, 1, 1, 0, 0, 0]
31: [1, 1, 1, 0, 0, 0]
37: [0, 0, 0, 0, 0, 1]
41: [2, 0, 0, 1, 0, 0]
43: [0, 0, 0, 0, 0, 1]
47: [0, 0, 0, 0, 0, 1]
53: [2, 0, 0, 1, 0, 0]
59: [2, 0, 0, 1, 0, 0]
61: [0, 0, 0, 0, 0, 1]
67: [1, 0, 0, 0, 1, 0]
71: [2, 0, 0, 1, 0, 0]
73: [0, 1, 0, 1, 0, 0]
79: [1, 1, 1, 0, 0, 0]
83: [1, 0, 0, 0, 1, 0]
89: [0, 1, 0, 1, 0, 0]
97: [3, 0, 1, 0, 0, 0]
```

```
X^7 + X^4 - 2X^2 + 8X + 4
 2: Not separable.
 3: Not separable.
 5: [0, 2, 1, 0, 0, 0, 0]
 7: [0, 2, 1, 0, 0, 0, 0]
11: [1, 3, 0, 0, 0, 0, 0]
13: [1, 3, 0, 0, 0, 0, 0]
17: [3, 2, 0, 0, 0, 0, 0]
19: [4, 0, 1, 0, 0, 0, 0]
23: [1, 3, 0, 0, 0, 0, 0]
29: [0, 2, 1, 0, 0, 0, 0]
31: [1, 3, 0, 0, 0, 0, 0]
37: [1, 3, 0, 0, 0, 0, 0]
41: [0, 2, 1, 0, 0, 0, 0]
43: [5, 1, 0, 0, 0, 0, 0]
47: [1, 3, 0, 0, 0, 0, 0]
53: [0, 2, 1, 0, 0, 0, 0]
59: Not separable.
61: [1, 3, 0, 0, 0, 0, 0]
67: [5, 1, 0, 0, 0, 0, 0]
71: [3, 2, 0, 0, 0, 0, 0]
73: [5, 1, 0, 0, 0, 0, 0]
79: [0, 2, 1, 0, 0, 0, 0]
83: [1, 3, 0, 0, 0, 0, 0]
89: [1, 3, 0, 0, 0, 0, 0]
97: [5, 1, 0, 0, 0, 0, 0]
```

```
X^7 - 2X^6 + 2X + 2
 2: Not separable.
 3: Not separable.
 5: [0, 0, 0, 0, 0, 0, 1]
 7: [0, 0, 0, 0, 0, 0, 1]
11: Not separable.
13: [0, 2, 1, 0, 0, 0, 0]
17: [1, 1, 0, 1, 0, 0, 0]
19: [1, 1, 0, 1, 0, 0, 0]
23: [0, 0, 0, 0, 0, 0, 1]
29: [0, 0, 0, 0, 0, 0, 1]
31: [0, 0, 0, 0, 0, 0, 1]
37: [0, 0, 0, 0, 0, 0, 1]
41: [2, 0, 0, 0, 1, 0, 0]
43: [2, 0, 0, 0, 1, 0, 0]
47: [0, 0, 0, 0, 0, 0, 1]
53: [1, 1, 0, 1, 0, 0, 0]
59: [0, 0, 0, 0, 0, 0, 1]
61: [1, 1, 0, 1, 0, 0, 0]
67: [2, 0, 0, 0, 1, 0, 0]
71: [0, 0, 0, 0, 0, 0, 1]
73: [1, 0, 2, 0, 0, 0, 0]
79: [1, 1, 0, 1, 0, 0, 0]
83: [0, 0, 0, 0, 0, 0, 1]
89: [2, 0, 0, 0, 1, 0, 0]
97: [1, 1, 0, 1, 0, 0, 0]
```

```
X^7 + X^5 - 4X^4 - X^3 + 5X +1
 2: [0, 0, 0, 0, 0, 0, 1]
 3: Not separable.
 5: [1, 3, 0, 0, 0, 0, 0]
 7: [0, 0, 0, 0, 0, 0, 1]
11: [1, 3, 0, 0, 0, 0, 0]
13: [1, 3, 0, 0, 0, 0, 0]
17: [0, 0, 0, 0, 0, 0, 1]
19: [0, 0, 0, 0, 0, 0, 1]
23: [1, 3, 0, 0, 0, 0, 0]
29: [0, 0, 0, 0, 0, 0, 1]
31: [0, 0, 0, 0, 0, 0, 1]
37: [0, 0, 0, 0, 0, 0, 1]
41: [0, 0, 0, 0, 0, 0, 1]
43: [0, 0, 0, 0, 0, 0, 1]
47: [0, 0, 0, 0, 0, 0, 1]
53: [0, 0, 0, 0, 0, 0, 1]
59: [1, 3, 0, 0, 0, 0, 0]
61: [1, 3, 0, 0, 0, 0, 0]
67: [1, 3, 0, 0, 0, 0, 0]
71: [1, 3, 0, 0, 0, 0, 0]
73: [0, 0, 0, 0, 0, 0, 1]
79: [1, 3, 0, 0, 0, 0, 0]
83: [0, 0, 0, 0, 0, 0, 1]
89: [0, 0, 0, 0, 0, 0, 1]
97: [1, 3, 0, 0, 0, 0, 0]
```

2 Analysis

Q4: Data. Our aim is to classify the Galois groups of the example polynomials f up to isomorphism. We can deduce from the above output that any cycle type listed for f appears as the cycle type of some element of $G(f)$. Thus, in finding $G(f)$, we only need to know which cycle types appear in the lists. Listed below are the cycle types yielded by `decomp`(f, p) at some point for p up to 2000.

```
X^2 + X + 41
    [0,1], [2]
X^3 + 2X + 1
    [1,1], [0,0,1], [3]
X^3 + X^2 - 2X - 1
    [0,0,1], [3]
X^4 - 2X^2 + 4
    [0,2], [4]

X^4 - X^3 - 4X + 16
    [0,0,0,1], [0,2], [2,1], [4]
X^4 - 2X^3 + 5X + 5
    [0,0,0,1], [1,0,1], [2,1], [0,2], [4]
X^4 + 7X^2 + 6X + 7
    [0,2], [4]
X^4 + 3X^3 - 6X^2 - 9X + 7
    [0,2], [2,1], [4]

X^5 + 36
    [1,0,0,1], [0,0,0,0,1], [1,2], [5]
X^5 - 5X + 3
    [0,1,1], [1,2], [2,0,1], [3,1], [5]
X^5 + X^3 - 3X^2 + 3
    [0,0,0,0,1], [2,0,1], [1,2], [5]
X^5 - 11X^3 + 22X - 11
    [0,0,0,0,1], [5]

X^6 + X + 1
    [0,0,0,0,0,1], [1,1,1], [0,0,2], [1,0,0,0,1], [0,1,0,1],
       [2,0,0,1], [3,0,1], [2,2], [0,3], [4,1]
X^7 - 2X^6 + 2X + 2
    [0,0,0,0,0,0,1], [0,2,1], [1,1,0,1], [2,0,0,0,1], [1,0,2],
       [3,2], [4,0,1]
X^7 + X^4 - 2X^2 + 8X + 4
    [0,2,1], [1,3], [3,2], [4,0,1], [5,1], [7]
X^7 + X^5 - 4X^4 - X^3 + 5X + 1
    [0,0,0,0,0,0,1], [1,3], [7]
```

Q4: Methods. Let's assemble a toolbox for our analysis. It will comprise 7 tools.

L: Lower bounds. The order of an element of a permutation group (like $G(f)$) is the lowest common multiple (LCM) of its cycle type, presented as an integer partition. Thus, every cycle type present in $G(f)$ determines an order present, which divides $|G(f)|$; hence, the

LCM of the LCM of the cycle types divides $|G(f)|$. Beyond that, since an element of order $n \in \mathbb{N}$ generates an embedding $C_n \hookrightarrow G(f)$, we get lower bounds on the number of elements in $G(f)$ of a certain order by counting them in C_n.

F: Full data. We can go further with our initial use of the data. In a way that will be made precise in the *Density* paragraph, each of our lists of cycle types is likely to comprise the full list of cycle types extant in $G(f)$. Note the obvious exceptions – polynomials 13 and 14 are missing the cycle type `[deg(f)]`, which corresponds to the identity of $G(f)$ and so must be there. We'll append those and assume that the resulting lists are indeed full.[3]

N: Irreducible degree factor lemma. Suppose f is irreducible. It has a root $\alpha \in F(f)$ when evaluated in $F(f)/\mathbb{Q}$ (its splitting field), and being monic, it is the minimal polynomial of α. We have the extensions $F(f)/\mathbb{Q}(\alpha)/\mathbb{Q}$, so $\deg(f) = [\mathbb{Q}(\alpha) : \mathbb{Q}] \big| [F(f) : \mathbb{Q}] = |G(f)|$, since $F(f)/\mathbb{Q}$ is a Galois extension (as a splitting field of an irreducible separable polynomial). Hence, $\deg(f) || G(f)|$, which tells us more than element orders if $\deg(f)$ is not prime.

P: Galois group product lemma. To deal with the occasional reducible polynomial,[4] we consider the structure of the Galois group of a product. Suppose $\exists p, q \in \mathbb{Q}[X]$ non-constant: $f = pq$. Let $\phi \in G(f)$. By commuting polynomial evaluation with ϕ (a restricted $\mathbb{Q}$-automorphism) and noting the injectivity of ϕ, we have $\phi(Z(p)) = Z(p)$ and $\phi(Z(q)) = Z(q)$. Restricting its domain and range thus yields $\phi_p \in G(p)$ and $\phi_q \in G(q)$. This gives rise to the homomorphism

$$G(f) = G(pq) \hookrightarrow G(p) \times G(q), \quad \phi \mapsto (\phi_p, \phi_q)$$

where homomorphism is immediate and injectivity follows because $Z(f) = Z(p) \cup Z(q)$. As p and q have lower degrees than f, their Galois groups are contained in much smaller symmetric groups, so this is a significant simplification.

Moreover, $\forall \phi \in G(f)$ $\phi = \tilde{\phi_p}\tilde{\phi_q}$, where $\tilde{\phi_p} \in G(f)$, $x \mapsto \phi_p(x)$ (if $x \in Z(p)$), x (else) (similarly for $\tilde{\phi_q}$), because f is separable, whence $Z(p) \cap Z(q) = \emptyset$. Hence, $G(f) \subseteq \{\tilde{\alpha}\tilde{\beta} : (\alpha, \beta) \in G(p) \times G(q)\}$, and since any $\tilde{\alpha}$, $\tilde{\beta}$ are disjoint and have the same cycle types as α, β (respectively), the cycle type of $\tilde{\alpha}\tilde{\beta}$ is their concatenation (as integer partitions), so we can eliminate elements whose cycle type is not present in $G(f)$.

I: Small index lemma. Suppose $n = \deg(f) \geq 5$, and $G(f) \leq \mathrm{Alt}(Z(f)) \cong A_n$. A_n is simple, so its proper subgroups H satisfy $\frac{n!}{2} = |A_n| \big| |A_n : H|$!.[5] Hence, either $G(f) = \mathrm{Alt}(Z(f))$ or $(n-1)! < \frac{n!}{2} \leq |\mathrm{Alt}(Z(f)) : G(f)|$!, whence $n \leq |\mathrm{Alt}(Z(f)) : G(f)|$, whence $|G(f)| \leq \frac{(n-1)!}{2}$.

$*$: $2/n - 1/n$ lemma. The younglings observed a hunched green figure drift into view.
"If a 2, an $n-1$, and an n cycle, you see, then the whole of S_n, it is."

S: Subgroup dossier. Only certain orders of group are permissible for subgroups of $S_{\deg(f)}$, and this is sometimes enough to determine the group once options have been narrowed down sufficiently. Since we are always looking for subgroups of S_n for some small $n \in \mathbb{N}$, we'll content ourselves to use tables that list the possibilities.[6]

[3] But, recognising the uncertainty here, we'll try to use this method as little as possible.

[4] We'll also presume we can determine reducibility and factorisations, for example by comparing coefficients and using bounding and divisibility in $\mathbb{Q}$ (or $\mathbb{Z}$, via Gauss's lemma).

[5] See *IB Groups, Rings and Modules* for a proof of this fact, which is a novel use of the left coset action.

[6] S_5 and S_7 tables attached.

Q4: Analysis. Let's get to work. The methods are referred to by code letter as they are applied, and each paragraph is numbered i according to the polynomial f_i it is discussing (in the order they are presented in the project). First, the easy ones: full symmetric groups.

1 $G \lesssim S_2$, and $|G| \geq 2$ (L), so $G \cong S_2$.

2 $G \lesssim S_3$, and $|G| \geq \mathrm{lcm}(2,3,1) = 6$, so $G \cong S_3$.

6 $G \cong S_4$ $(*)$.

13 $G \cong S_6$ $(*)$.

Next, some reducible polynomials. Notice that for any irreducible quadratic polynomial f, $G(f) = \mathrm{Sym}(Z(f))$, since $2||G|$ (N).

8 f has irreducible factorisation[7] $(X^2+X-1)(X^2+2X-7)$, so $G \lesssim C_2^2$ (P). $|G| \geq 3$ (L – there is at least 1 element of each of 3 cycle types), so $|G| = 4$, so $G \cong C_2^2$.

10 f has irreducible factorisation $(X^3-X^2+2X-3)(X^2+X-1)$. Computing decompositions of the cubic factor, we find:

```
>>> cycles([-3,2,-1,1],1,2000)[0]
[[0, 0, 1], [1, 1], [3]]
```

so its Galois group $G \lesssim S_3$ has order $\geq \mathrm{lcm}(3,2,1) = 6$, so is S_3. The quadratic has Galois group $\cong C_2$. Hence, $G \lesssim S_3 \times C_2$ (P). $|G| \geq 7$ (L – at least 2 elements of type `[0,1,1]`, 1 of type `[1,2]`, 2 of type `[2,0,1]`, 1 of type `[3,1]`, 1 of type `[5]`), so $|G| = 12$, so $G \cong S_3 \times C_2$.

7 f has irreducible factorisation pq, where $p = (X^2 - X + 7)$ and $q = (X^2 + X + 1)$. This time, we need to consider the Galois group concretely to get enough information. Let $Z(p) = \{a, b\}$, $Z(q) = \{c, d\}$. Then $G(p) = \{\iota, (a\ b)\}$ and $G(q) = \{\iota, (c\ d)\}$, so $G(f) \subseteq \{\iota, (a\ b), (c\ d), (a\ b)(c\ d)\}$ (P). G contains no element of type `[2,1]` (M), so $G(f) \subseteq \{\iota, (a\ b)(c\ d)\}$. $|G| \geq 2$ (L), so $G \cong C_2$.

Polynomial 7 shows that irreducibility is necessary in (N) and provides a counterexample to the conjecture that the embedding $G(pq) \hookrightarrow G(p) \times G(q)$ in (P) is always an isomorphism. Manual work shows that, when the splitting fields are constructed in $\mathbb{C}/\mathbb{Q}$ (algebraically closed),

$$Z(p) = \left\{\frac{1}{2} \pm \frac{3}{2}i\sqrt{3}\right\} \qquad Z(q) = \left\{-\frac{1}{2} \pm \frac{1}{2}i\sqrt{3}\right\}$$

whence $F(f) = F(p) = F(q) = \mathbb{Q}(\sqrt{3}i) \supsetneq \mathbb{Q}$. Thus, the degeneracy manifests itself as the fact that $(F(p) \cap F(q)) \backslash \mathbb{Q} \neq \emptyset$. We cannot transpose the roots of p or q without permuting the whole splitting field, which also transposes the roots of the other factor.

Polynomial 7 is also our first application of (F), which turns out to be very useful. It allows us to discount elements of particular orders (thus precluding certain prime divisors of G) or prove that $G \leq \mathrm{Alt}(Z(f)) \cong A_{\deg(f)}$ (if all cycle types are even).

4 f is irreducible, so $4||G|$ (N). $G \lesssim A_4$ (M), so $|G| = 4, 12$. G has no order 3 or 4 element (M), so $3 \nmid |G|$ and $G \ncong C_4$, so $G \cong C_2^2$.

12 $G \lesssim S_5$, G has no order 2 or 3 element (M) and $5||G|$ (L), so $|G| = 5$, so $G \cong C_5$.

[7] With monic representatives of the associate classes of irreducibles.

If we are not aiming to prove that $G(f)$ is a product of symmetric groups corresponding to its irreducible factors, we have no way in our toolbox except (M) of obtaining an upper bound on its size, so we can't guarantee our deductions. To do this, we could resort to manually computing splitting fields, or use the fact[8] that $G(f) \leq \mathrm{Alt}(Z(f)) \iff D_f$ is square, where D_f is the discriminant of f. However, for large $\deg(f)$, this is too expensive to do by hand.

Next, we find some full alternating groups; in these examples, the use of (M) could be replaced by computing discriminants using a computer.

3 $G \lesssim A_3$ (M). $|G| \geq 3 = |A_3|$ (L), so $G \cong A_3 \cong C_3$.

11 $G \lesssim A_5$ (M). $|G| \geq 30$ (L), so $G \cong A_5$ (I – else $|G| \leq \frac{4!}{2} = 12$, contradiction).

14 $G \lesssim A_7$ (M). $|G| \geq 420$ (L), so $G \cong A_7$ (I – else $|G| \leq \frac{6!}{2} = 360$, contradiction).

To find more exotic Galois groups, consulting subgroup lists (S) is fastest.

5 $G \lesssim S_4$. G has no order 3 element (M), so $|G| = 1, 2, 4, 8$. $|G| \geq 5$ (L – at least 2 elements of type `[0,0,0,1]`, and 1 of the other 3 types), so $|G| = 8$, so $G \cong D_8$ (S).

9 $G \lesssim S_5$. G has no order 3 element (M), so $|G| = 1, 2, 4, 5, 8, 10, 20, 40$. $|G| \geq 20$ (L), so $|G| = 20, 40$, so $G \cong GA(1,5)$ (S) (and $|G| = 20$).[9]

16 $G \lesssim S_7$. G has no order 3,5 element (M), so $|G| = 1, 2, 4, 7, 8, 14, 16, 28, 56, 112$. $14||G|$ (L), so $|G| = 14, 28, 56, 112$, so $G \cong D_{14}$ (S) (and $|G| = 14$).

One polynomial remains.

15 f_{15} has irreducible factorisation $f_4 f_2 ...$ Let $Z(f_2) = \{x, y, z\}$ and $Z(f_4) = \{a, b, c, d\}$. Then $G(f_2) = \mathrm{Sym}(Z(f_2))$ and $G(f_4) = \{\iota, (a\ b)(c\ d), (a\ c)(b\ d), (a\ d)(b\ c)\}$ (we know that $G(f_4)\backslash\{\iota\}$ consists only of double transpositions, for which these are the only options). Hence, $G(f_{15}) \subseteq \{\tilde{\alpha}\tilde{\beta} : (\alpha, \beta) \in G(f_2) \times G(f_4)\}$ (P). $G(f_{15})$ contains elements of types `[4,0,1]` and `[5,1]`, so $\{\tilde{\alpha} : \alpha \in G(f_2)\} \subseteq G_(f_{15})$. It also contains an element of type `[3,2]`, which must be $\tilde{\beta}$ for some $\beta \in G(f_4)\backslash\{\iota\}$, w.l.o.g. $(a\ b)(c\ d)$. Thus, there are two possibilities:

$$G(f_{15}) = \begin{cases} \{\tilde{\alpha}\tilde{\beta} : (\alpha, \beta) \in G(f_2) \times G(f_4)\} & \cong S_3 \times C_2^2 \\ \{\tilde{\alpha}\tilde{\beta} : (\alpha, \beta) \in G(f_2) \times \{\iota, (a\ b)(c\ d)\}\} & \cong S_3 \times C_2 \end{cases}$$

It seems like we've taken our methods as far as they'll go, but cannot distinguish the two.

Q4: Density. Surprisingly enough, when we summarised which cycle types appeared in decompositions of each polynomial f, we actually lost information. We could also keep the relative frequency of a given cycle type appearing in decompositions w.r.t. the primes up to $n \in \mathbb{N}$ for which the decomposed polynomial is separable. This is implemented as `freq`(f, n). With $n = 2000$, this is the output of `freq` on the first 4 example polynomials.

```
1:  [0,1]   [2]                    3:  [0,0,1]   [3]
    0.540   0.460                      0.679     0.321
2:  [1,1]   [0,0,1]   [3]          4:  [0,2]   [4]
    0.513   0.338     0.149            0.777   0.223
```

[8] See theorem 4.7 in *Galois Groups as Permutation Groups*, Keith Conrad.

[9] This is the *general affine group of degree 1 over* $\mathbb{F}_5$.

This is a noisy but nonetheless obvious copy of another pattern. Since $G(f)$ is, for the first 4 polynomials, always a full symmetric or alternating group, or has only one cycle type corresponding to each order of an element, it is easy to calculate the proportion of its elements that have a given cycle type. The results are as follows:

```
1:  [0,1]   [2]                      3:  [0,0,1]   [3]
    0.500   0.500                        0.667     0.333
2:  [1,1]   [0,0,1]   [3]            4:  [0,2]   [4]
    0.500   0.333     0.167              0.750   0.250
```

Theorem (Tschebotareff Density Theorem)**.** [10]

Let $f \in \mathbb{Z}[X]$. Let σ be a cycle type of an element of $S_{\deg(f)}$. Then as $n \to \infty$,

$$\frac{|\{p \in \mathbb{P} \cap [1,n] : \textit{the decomposition of } f \bmod p \textit{ yields } \sigma\}|}{|\mathbb{P} \cap [1,n]|} \to \frac{|g \in G(f) : g \textit{ has cycle type } \sigma|}{|G(f)|}$$

The convergence appears to be quite slow; in particular, finding the relative frequencies for primes up to 100 can give very misleading results (not least because some cycle types are more likely to not show up at all). This computation is shown below – note that, if we round each frequency to the nearest multiple of $1/|G|$, we would get incorrect results for polynomial 2. Even taking into account the presence of the type-`[3]` element, it's not clear whether there is one less element of type `[1,1]` or of type `[0,0,1]`.

```
1:  [0,1]   [2]                      3:  [0,0,1]   [3]
    0.692   0.308                        0.720     0.280
2:  [1,1]   [0,0,1]   [3]            4:  [0,2]   [4]
    0.600   0.320     0.080              0.792   0.208
```

Let's substantiate the theorem with more empirical evidence from example polynomials f for which the order distribution of $G(f)$ is easy to calculate (in the case of reducibles, we use (P) to do this). The first row of data is the empirical result of `freq`$(f, 2000)$; the second is the order proportion.

```
 6:  [0,0,0,1]   [1,0,1]   [2,1]   [0,2]   [4]
     0.267       0.317     0.257   0.123   0.037
     0.250       0.333     0.250   0.125   0.042

 7:  [0,2]   [4]
     0.510   0.490
     0.500   0.500

 8:  [0,2]   [2,1]   [4]
     0.263   0.507   0.230
     0.250   0.500   0.250

10:  [0,1,1]   [1,2]   [2,0,1]   [3,1]   [5]
     0.173     0.266   0.163     0.336   0.063
     0.167     0.250   0.167     0.333   0.083

11:  [0,0,0,0,1]   [2,0,1]   [1,2]   [5]
     0.403         0.367     0.220   0.010
     0.400         0.333     0.250   0.017
```

[10] See *Techniques for the Computation of Galois Groups*, Alexander Hulpke, for a statement and reference.

```
12: [0,0,0,0,1]   [5]
    0.817         0.183
    0.800         0.200

13: [0,0,0,0,0,1]   [1,1,1]   [0,0,2]   [1,0,0,0,1]   [0,1,0,1]
    0.229           0.203     0.040     0.183         0.093
    0.167           0.167     0.056     0.200         0.125
    [2,0,0,1]   [3,0,1]   [2,2]   [0,3]   [4,1]
    0.123       0.066     0.037   0.010   0.017
    0.125       0.056     0.063   0.021   0.021

14: [0,0,0,0,0,0,1]   [0,2,1]   [1,1,0,1]   [2,0,0,0,1]   [1,0,2]
    0.300             0.090     0.220       0.230         0.103
    0.286             0.083     0.250       0.200         0.111
    [3,2]   [4,0,1]
    0.033   0.023
    0.042   0.028
```

The data correlates reasonably well. For polynomials 11, 13, 14 (with Galois groups of order larger than 24), however, we can already see that rounding to the nearest multiple of $\frac{1}{|G|}$ yields inaccuracies, and indeed, 13 and 14 are missing the cycle type `[deg(`f`)]`.

This type corresponds only to the identity, and thus has asymptotic frequency $\frac{1}{|G|}$ – the smallest possible. Hence, it is no surprise that the identity is missing from the decompositions of the polynomials with the largest groups. We expect it to appear in the decompositions of the first $|G|$ primes – i.e. for primes up to 5443 (polynomial 13) and 22543 (polynomial 14).[11] This qualitatively goes some way towards justifying the full data assumption (F).

For polynomials f for which it is harder to calculate the distribution of cycle types in $G(f)$ manually, we estimate them computing decomposition frequencies (again with primes up to 2000) and rounding them to the nearest multiple of $\frac{1}{|G|}$.

```
 5: [0,0,0,1]   [0,2]   [2,1]   [4]
    0.257       0.380   0.260   0.103
    0.250       0.375   0.250   0.125

 9: [1,0,0,1]   [0,0,0,0,1]   [1,2]   [5]
    0.513       0.190         0.243   0.053
    0.500       0.200         0.250   0.050

16: [0,0,0,0,0,0,1]   [1,3]   [7]
    0.395             0.548   0.056
    0.429             0.571   0.071
```

The last of these is clearly erroneous since the proportions sum to more than 1... however, we know the number of order 7 elements is divisible by 6 (partitioning them by the subgroups they generate), so is 6 or 12. The data suggests it *is* 6, so we can infer (since there is one type-`[7]` element) that there are very likely 7 type-`[1,3]` elements – i.e. 0.571 should be 0.500.

Finally, what can we say about polynomial 15?[12] We have two hypotheses on its isomorphism class, and know the cycle type frequencies in both cases. The first two rows of data below list them. We also have decomposition frequencies for primes up to 2000 (third row).

[11] It turns out that the smallest prime that yields it is 4339 for 13 and 60139(!) for 14.

[12] The polynomial that lived.

```
                [0,2,1]   [1,3]   [3,2]   [4,0,1]   [5,1]   [7]
S3 x C_2^2:     0.250     0.375   0.125   0.083     0.125   0.042
  S3 x C_2:     0.167     0.250   0.083   0.167     0.250   0.083
 Predicted:     0.271     0.381   0.127   0.067     0.130   0.023
```

The choice is clear. Hence, assuming the accuracy of this data, we have determined that $G(f_{15}) \cong S3 \times C_2^2$, which completes the set. To celebrate, here's a list of the (isomorphism) classes and orders of the Galois groups of the example polynomials.

i	G	$\lvert G\rvert$	i	G	$\lvert G\rvert$	i	G	$\lvert G\rvert$	i	G	$\lvert G\rvert$
1	S_2	2	*5*	D_8	8	*9*	$GA(1,5)$	20	*13*	S_6	720
2	S_3	6	*6*	S_4	24	*10*	$S_3 \times C_2$	12	*14*	A_7	2520
3	A_3	3	*7*	C_2	2	*11*	A_5	60	*15*	$S_3 \times C_2^2$	24
4	C_2^2	4	*8*	C_2^2	4	*12*	C_5	5	*16*	D_{14}	14

A Programs

The programs take the form of three modules: `C.py`, `Poly.py`, `Galois.py`. Program output printed in the report is generated by functions in `Output.py`. `InitC.py` is a script that, when run in a Python shell, loads all functions into the global memory.

A.1 Documentation

This section summarises the purpose of the project's more primitive functions.

`rf(n)`: $n \in \mathbb{N}_0$. Returns $\sqrt{\lfloor N \rfloor}$.

`primes(m,n)`: $m, n \in \mathbb{Z}$. Yields the next prime number each time it's called,[13] starting from m and ending at n.

`modinv(a,p)`: $p \in \mathbb{P}$, $a \in \mathbb{Z}_p$. Returns $a^{-1} \bmod p$ if $a \neq 0$.

`c(a)`: a a list. Deletes trailing zeros from a.

`pj(a,p)`: $a \in \mathbb{Z}[X]$, $p \in \mathbb{P}$. Projects a into $\mathbb{Z}_p[X]$.

`add(a,b,p)`: $a, b \in \mathbb{Z}[X]$, $p \in \mathbb{P}$. Returns $a + b$ projected into $\mathbb{Z}_p[X]$.

`mp(a,b,p)`: $a, b \in \mathbb{Z}[X]$, $p \in \mathbb{P}$. Returns ab projected into $\mathbb{Z}_p[X]$.

`sc(a,s,p)`: $a \in \mathbb{Z}[X]$, $s \in \mathbb{Z}$, $p \in \mathbb{P}$. Returns sa projected into $\mathbb{Z}_p[X]$.

Remaining functions: Documented within the report itself.

[13] Artist's impression: http://tinyurl.com/26z6zo.

A.2 C

```
def rf(n):
    i = 1; x = 0
    while True:
        if (x + i) ** 2 <= n:
            i <<= 1
        elif i != 1:
            x += i >> 1; i = 1
        else:
            return x

def primes(m,n):
    m = max(m,2)
    while True:
        for r in range(2,rf(m)+1):
            if m % r == 0:
                break
        else:
            yield m
        if m > n:
            break
        else:
            m += 1

def modinv(a,p):
    a %= p
    if a == 0: print('0 not invertible.'); return
    x = 1
    for i in range(p-2):
        x *= a; x %= p
    return x
```

A.3 Poly

```
from C import *
from itertools import zip_longest as zipl

def c(a):
    for i in range(len(a)-1,-1,-1):
        if a[i] == 0: a.pop()
        else: break
    return a

def pj(a,p):
    return c([x % p for x in a])

def add(a,b,p):
    return pj([x[0]+x[1] for x in zipl(a,b,fillvalue=0)],p)

def mp(a,b,p):
    return pj([sum([(a[j]*b[i-j])%p for j in range(i+1) if j<len
       (a) and i-j<len(b)]) for i in range(len(a)+len(b)-1)],p)

def sc(a,s,p):
    return pj([s*x for x in a],p)

def div(a,b,p):
    if len(b) == 0: print('Div/0.'); return
    t = modinv(b[-1],p); b = sc(b,t,p)
    n = len(a) - len(b)
    q = [0 for i in range(n+1)]
    while n >= 0:
        q[n] = a[-1]
        a = add(a,sc(([0 for i in range(n)] + sc(b,a[-1],p)),-1,
           p),p)
        n = len(a) - len(b)
    return [sc(q,t,p),a]

def hcf(a,b,p):
    return sc(a,modinv(a[-1],p),p) if len(b) == 0 else hcf(b,div
       (a,b,p)[1],p)

def exp(a,b,n,p):
    mpa = lambda r: div(mp(r,a,p),b,p)[1]
    x = [1]
    while n != 0:
        if n & 1: x = mpa(x)
        a = mpa(a)
        n >>= 1
    return x
```

A.4 Galois

```
from C import *
from Poly import *

def decomp(f,p):
    f = pj(f,p)
    d = lambda a,p: pj([i*a[i] for i in range(1,len(a))],p)
    if len(hcf(f,d(f,p),p)) != 1:
        return 'Not separable.'
    a = [0]
    for r in range(1,len(f)):
        m = add(exp([0,1],f,p**r,p),div([0,p-1],f,p)[1],p)
        a.append(hcf(f,m,p))
        f = div(f,a[r],p)[0]
    return [(len(a[i])-1)//i for i in range(1,len(a))]

def cycles(f,m,n):
    x = []; y = []
    for p in primes(m,n):
        a = decomp(f,p)
        if a not in x and a != 'Not separable.':
            x.append(a); y.append(p)
    return [[c(t) for t in x],y]

def freq(f,n):
    x = []; y = []; j = 0
    for p in primes(1,n):
        r = decomp(f,p)
        if r != 'Not separable.':
            j += 1
            if r in x:
                y[x.index(r)] += 1
            else:
                x.append(r); y.append(1)
    return [[c(i) for i in x],[round(i/j,3) for i in y]]
```