In [1]:

```
import pandas as pd
from sympy.polys.domains import ZZ
from sympy.polys.galoistools import gf_frobenius_monomial_base, gf_sub, gf_gcd, g
f_frobenius_map
from random import randrange
from sympy.ntheory import factorint
from sympy.polys.polytools import invert
from functools import reduce


data = pd.read_csv('ieeja.txt', header=None)
data.columns = ['number']
number_list = data['number'].values.tolist()
p = None


def galois_field_monic(f, p):
    lc = f[0]

    if lc == 1:
        return list(f)

    a = invert(lc, p)
    return [ (a*b) % p for b in f ]

def is_polynomial_irreducible(f, p):
    """
    Rabin's polynomial irreducibility test over finite fields.
    """
    gf_degree = len(f) - 1

    if gf_degree <= 1:
        return True

    monic_field = galois_field_monic(f, p)

    compare_polynomial = [1, 0]

    indices = { gf_degree//i for i in factorint(gf_degree) }

    monomial_base = gf_frobenius_monomial_base(monic_field, p, ZZ)
    h = monomial_base[1]

    for i in range(1, gf_degree):
        if i in indices:
            g = gf_sub(h, compare_polynomial, p, ZZ)

            if gf_gcd(monic_field, g, p, ZZ) != [1]:
                return False

        h = gf_frobenius_map(h, monic_field, monomial_base, p, ZZ)

    return h == compare_polynomial

def generate_irreducible_polynomial_over_field(d, p):
    """
    Basic algorithm:

        1. Generate random polynomial over finite field Zp and degree d
        2. Check if polynomial is irreducible
        3. Return irreducible polynomial
    """
    while True:
        random_polynomial = [1] + [randrange(p) for _ in range(d)]
        try:
            if is_polynomial_irreducible(random_polynomial, p):
                return random_polynomial
        except:
            continue
```

```
def format_polynomial(power: int, multiply: int) -> str:
    if power == 0:
        return "{multiply}".format(multiply=multiply)
    if multiply == 0:
        return ""
    if multiply == 1:
        return "x^{power}".format(power=power)
    return "{multiply}*x^{power}".format(multiply=multiply, power=power)

polynomial_list = []
for d in number_list:
    if not p:
        p = d
        continue

    irreducible_polynomial = generate_irreducible_polynomial_over_field(d, p)
    polynomial = [format_polynomial(i, x) for i, x in enumerate(reversed(irreduci
ble_polynomial))]
    polynomial_string = " + ".join(reversed(polynomial))

    polynomial_list.append({"polynomial": polynomial_string, "d": d, "p": p})

    p = d


for p in polynomial_list:
    print("Moniska nereducējama {d}-tās pakāpes polinomu pār lauku Z_{p}: {polyno
mial}".format(
        polynomial=p["polynomial"],
        d=p["d"],
        p=p["p"]
    ))
```

```
Moniska nereducējama 5-tās pakāpes polinomu pār lauku Z_7: x^5 + 5*x^4 + x^3 + 5
*x^2 + 4*x^1 + 1
```