```
34 usages
public class Poly {
    28 usages
    private int[] coef;
    21 usages
    private int deg;
    1 usage
    private int passedcoef;
    1 usage
    private int passedexp;

    10 usages
    public Poly(int m, int n) {
        passedcoef = m;
        passedexp = n;

        coef = new int[n+1];
        coef[n] = m;
        deg = degree();
    }
```

```
    4 usages
    public int degree() {
        int d = 0;
        for (int i=0; i<coef.length; i++)
            if (coef[i] != 0) d = i;
        return d;
    }

    public int coeff(int d) { return coef[d]; }

    4 usages
    public Poly add(Poly p) {
        Poly current = this;
        Poly answer = new Poly( m: 0, Math.max(current.deg, p.deg));
        for (int i=0; i<=current.deg; i++) answer.coef[i] += current.coef[i];
        for (int i=0; i<=p.deg; i++) answer.coef[i] += p.coef[i];
        answer.deg = answer.degree();
        return answer;
    }
```

```
5 usages
public Poly sub(Poly p) {
    Poly current = this;
    Poly answer = new Poly( m: 0, Math.max(current.deg, p.deg));
    for (int i=0; i<=current.deg; i++) answer.coef[i] += current.coef[i];
    for (int i=0; i<=p.deg; i++) answer.coef[i] -= p.coef[i];
    answer.deg = answer.degree();
    return answer;
}

1 usage
public Poly mult(Poly p) {
    Poly current = this;
    Poly answer = new Poly( m: 0,  n: current.deg + p.deg);
    for (int i=0; i<=current.deg; i++)
        for (int j=0; j <= p.deg; j++)
            answer.coef[i+j] += (current.coef[i]*p.coef[j]);
    answer.deg = answer.degree();
    return answer;
}
```

```
2 usages
public Poly minus() {
    Poly current = this;
    for (int i=0; i<coef.length; i++) {
        coef[i] = 0-coef[i];
    }
    return this;
}


public String toString() {
    if (deg == 0) return "" + coef[ 0 ];
    if (deg == 1) return coef[1] + "x + " + coef[0];
    String s = coef[deg] + "x^" + deg;
    for (int i = deg-1; i>=0; i--){
        if (coef[i] == 0){
            continue;
        } else if(coef[i] > 0) {
            s = s + " + " + (coef[i]);
        } else if(coef[i] < 0) s = s + " - " + (-coef[i]);
        if (i == 1) {
            s = s + "x";
        } else if (i > 1) s = s + "x^" + i;
    }
    return s;
}
```

```
    public static void main(String[] args) {
        Poly zero = new Poly( m: 0,  n: 0);

        Poly p1 = new Poly( m: 1, n: 2);
        Poly p2 = new Poly( m: 3,  n: 7);
        Poly p3 = new Poly( m: -5,  n: 0);
        Poly p4 = new Poly( m: 0,  n: 1);

        Poly q1 = new Poly(  m: 1,  n: 1 );
        Poly q2 = new Poly(  m: 3,  n: 0 );
        Poly q = q1.add(q2);

        Poly p = p1.add(q);
        Poly p5 = p1.add(p1);
        System.out.println(p);
        System.out.println(p5);

        System.out.println(zero);
        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p1.add(p3));
        System.out.println(p1.sub(p2));

        System.out.println(p1.sub(p2).sub(p3).sub(p4).sub(p1));
```

```
        System.out.println(p1.mult(p2));
        System.out.println(p2.minus());
        System.out.println(p.minus());
    }

}
```