Exercise: polynomial evaluation.

for $f \in \mathbb{Z}[x]$, how to represent f in c++?

Say $f(x) = \sum_{i=0}^{d} c_i x^i$.

Could store f by keeping the $\{c_i\}_{i=0}^{d}$ in an array or vector C.

$(C[i] \equiv c_i)$

Exercise: write a function for evaluating a polynomial at a given input.

```
int polyEval(const vector<int>& C, int x)
{   // need to output f(x) = sum_{i=0}^{C.size()-1} C[i] * x^i
    int sum = 0;
    for (int i=0; i<C.size(); i++) {
        sum += C[i] * pow(x,i);
    }
    return sum;
}

// How many multiplications required?
// Say n = C.size().
// 1 + 2 + 3 + ... + n = n(n+1)/2 > (1/2) n^2
```

Observation: pow(x,i) re-does a lot of

$1 + \frac{1}{\approx i}$

Work! $x^{i-1} \xrightarrow[*x]{} x^i$

pow(·,·) has no memory ...

```
int polyEval(const vector<int>& C, int x)
{   // need to output f(x) = sum_{i=0}^{C.size()-1} C[i]*x^i
    int sum = 0;
    int xi = 1;   // stores x^i
    for (int i=0; i<C.size(); i++) {
        sum += C[i] * xi;
        xi *= x;
    }
    return sum;
}
```

// Now how many multiplications (if $n = C.size()$)?

// $2n$. $2n \ll \frac{1}{2}n^2 < \frac{n(n+1)}{2}$

for large n.

Can we do even better??

Say $f(x) = 1 + 3x + 2x^2 + 4x^3$ $(C = [1, 3, 2, 4])$

4

↳ $4x+2$ // mult. partial result by x, // then add next coeff.

↳ $x(4x+2)+3$

↳ $x(4x^2+2x+3)+1$ ✓

```
int polyEval(const vector<int>& C, int x)
{
    // need to output f(x) = sum_{i=0}^{C.size()-1} C[i] * x^i
    int sum = C[C.size()-1];
    for (int i = C.size()-2; i >= 0; i--) {
        sum = sum * x + C[i];
    }
    return sum;
}

// Now # multiplications is only ≈ n.
```

yay!