More exercises w/ arrays.

Polynomial evaluation:

$$f(x) = \sum_{i=0}^{d} a_i x^i$$

input: f, x   output f(x)

How to store/represent f? Could use an array/vector of coefficients.

```
int eval(const vector<int>& a, int x) {
    // (int* a, int d, int x) if using arrays.
    // very similar to computing a sum...
    int sum = 0;
    int lastxi = 1;
    for (size_t i = 0; i < a.size(); i++) {
        // sum += a[i] * pow(x,i);
        // above takes too long. Throws away
        // useful work (x·x^{i-1} = x^i)
        sum += a[i] * lastxi;
        lastxi *= x;
    }
    return sum;
}
```

Note: above takes 2 * a.size() multiplications. Can we do better??

a_d

$\hookrightarrow a_d * x + a_{d-1}$

$\hookrightarrow (a_d * x + a_{d-1}) * x + a_{d-2}$

⋮

The # of factors of x that a coeff. has will be equal to the # of steps the coeff. has been in the accumulator. E.g., a_d will be multiplied by x^d. (This trick is called "Horner's Rule".)

```
int eval (const vector<int>& a, int x) {
     int sum = a[a.size()-1];
     for (size_t i = a.size()-2; i != (size_t)-1; i--) {
          sum = sum*x + a[i];
     }
     return sum;
}
```

Yay! only takes d multiplications
(a.size()-1).