Project 4

You are going to write a complete C program which implements the following functionality:

- Your program will read two input files:
  - values.txt
  - polynomial.txt
- Your program will create a file:
  - evaluations.txt
- Your program will evaluate the same polynomial for each value read from values.txt and write the results to evaluations.txt

values.txt

This file holds double numbers separated by whitespace.

```
12.5 5 67.89 -6 -13.37
```

There may be as many as `100` double numbers in this file.

polynomial.txt

This file holds a polynomial in a character array form.

```
5x+23.5x^3-x^2
```

There will only be one polynomial expression. monomials are not ordered according to the powers of the variable x. The coefficient of x at each monomial is written before the character `x`. Powers of x is represented by character `^` followed by a number. Each monomial will certainly include a character `x`.

The length of a polynomial expression can reach up to `1000` characters.

evaluations.txt

This file will hold the results of polynomial evaluations for each value read from `values.txt`. If your polynomial string is `5x+23.5x^3-x^2`, set `x` to be the value(one of the numbers read from values.txt) and evaluate the mathematical expression: `evaluation = 5*x + 23.5*x*x*x - x*x`. For the given example above, evaluations.txt will be as follows:

```
45804.69
2937.50
7349081.25
-5142.00
-56410.13
```

Remarks:

- In order to convert char arrays to numbers, you can use function `sscanf()` which is defined in `<stdio.h>`. For example:

```
double d1,d2;
char a[] = "12.5 63.4"
sscanf(a, "%lf%lf", &d1, &d2);
/* d1 stores 12.5 and d2 stores 63.4 */
```

- In order to find powers of a number, you can use `pow()` function defined in `<math.h>`

Turn in:

A complete C program `<Project 4.c>` which can be compiled using the following command:

```
gcc -std=c99 Project 4.c -o Project4
```