Homework #1

A program for finding roots of equations

Requirements

- Write a program in ***C*** that works as follows:

- The initial screen of the program

```
[ ID : your student ID ]
[ Name : your name ]

0. Input a polynomial equation

1. Find a root using Bisection
2. Find a root using False-position
3. Find a root using Fixed-point iteration
4. Find a root using Newton-Raphson

5. Quit

>
```

0. Input a polynomial equation

- Allow the user to input a ***polynomial*** equation
  (1) Receive its order ($2 \le \text{order} \le 10$)
  (2) Receive its coefficients

- Example

```
> Order: 3
> Coefficients: 1 1 1 1                    // x^3 + x^2 + x + 1 = 0
```

```
> Order: 3
> Coefficients: 4 0 1 2                    // 4x^3 + x + 2 = 0
```

```
> Order: 2
> Coefficients: 2 -0.4 -1                  // 2x^2 – 0.4x – 1 = 0
```

```
> Order: 4
> Coefficients: 0.4 0.5 0 9.4 -2.4         // 0.4x^4 + 0.5x^3 + 9.4x – 2.4 = 0
```

1. Find a root using Bisection

- Find a root of the input equation using ***the bisection method*** and display the result
  - If the user presses any key, return back to the initial screen

- Example

```
> xl = 0
> xu = 1
> es = 0.005          // 0.5%
> imax = 10000

Found root = 0.5647
```

2. Find a root using False-position

- Find a root of the input equation using ***the false-position method*** and display the result
  - If the user presses any key, return back to the initial screen

- Example

```
> xl = 0
> xu = 1.3
> es = 0.02          // 2%
> imax = 10000

Found root = 1.0156
```

3. Find a root using Fixed-Point Iteration

- Find a root of the input equation using ***fixed-point iteration*** and display the result
  - If the user presses any key, return back to the initial screen

- Example

```
> x0 = 0
> es = 0.01          // 1%
> imax = 10000

Found root = 0.5648
```

4. Find a root using Newton-Raphson

- Find a root of the input equation using ***the Newton-Raphson method*** and display the result
  - If the user presses any key, return back to the initial screen

- Example

```
> x0 = 0
> es = 0.02          // 2%
> imax = 10000

Found root = 0.7982
```

5. Quit

- Exit the program

Submission (1/2)

- Zip the following files into *studentID.zip* (e.g., 08xxxxx.zip)
  - Source file(s) (`*.c`, `*.h`)
    - Including related files required to compile and run the program (if any)
      - (ex) `Makefile`
  - Report.txt
    - Briefly describe how to compile and run your program
    - If you have any assumption, state it clearly

- Upload the Zip file to the e-class board
  - E-class → [Homework] → [HW#1]

Submission (2/2)

- Due
  - 2011.11.14 (Mon) 24:00
    - 1-day delay: 80% credit
    - 2-day delay or more: 0% credit

- Do not copy someone else's homework or let someone copy yours
  - You will get a ***zero*** for the assignment

Tips (1/3)

- You will need to evaluate $f(x)$ for a given x
  - ***Note that the following is just a sketch!***

```
int f_order;
float f_coefficients[10];

float evaluate_f(float x)
{
    float value = 0.0;

    for (i = order; i >= 0; i--)
        value += coefficients[i]*pow(x, order);

    return value;
}
```

Tips (2/3)

- To implement fixed-point iteration, you will need to obtain $g(x)$ for a given $f(x)$
  - ***Note that the following is just a sketch!***

```
int f_order;
float f_coefficients[10];

int g_order;
float g_coefficients[10];

void obtain_g()
{
    g_order = f_order;

    for (i = g_order; i >= 0; i--)
        g_coefficients[i] = -1 * f_coefficients[i];

    g_coefficients[1]++;
}
```

Tips (3/3)

- To implement Newton-Raphson, you will need to obtain $f'(x)$ for a given $f(x)$
  - ***Note that the following is just a sketch!***

```
int f_order;
float f_coefficients[10];

int f'_order;
Float f'_coefficients[10];

void obtain_f'()
{
    f'_order = f_order – 1;

    for (i = f'_order; i >= 0; i--)
        f'_coefficients[i] = (i + 1) * f_coefficients[i + 1];
}
```