Programming Refresher Workshop

Session 2 Exercises

Learning objectives:

- Using selection and repetition statements
- Writing functions/methods
- Applying neat logic in problem solving

Exercise 6 (ex6): Bisection Method

[Past CS1010 lab exercise.]
Numerical analysis is an important area in computing. One simple numerical method we shall study here is the *Bisection method*, which computes the root of a continuous function. The **root**, r, of a continuous function f is a value such that $f(r) = 0$.

How does bisection method work? It is best explained with an example. Given this polynomial function

$$p(x) = x^3 + 2x^2 + 5$$

we need to first provide two **endpoints** a and b such that the signs of $p(a)$ and $p(b)$ are different. For example, let $a = -3$ (hence $p(a) = -4$) and $b = 0$ (hence $p(b) = 5$). Here, the signs of $p(a)$ and $p(b)$ are different.

The principle is that, the root of the polynomial (that is, the value r where $p(r) = 0$) must lie somewhere between a and b. So for the above polynomial, the root r must lie somewhere between –3 and 0, because $p(-3)$ and $p(0)$ have opposite signs. (NOT because –3 and 0 have opposite signs!)

This is achieved as follows. The bisection method finds the midpoint m of the two endpoints a and b, and depending on the sign of $p(m)$ (the function value at m), it replaces either a or b with m (so m now becomes one of the two endpoints). It repeats this process and stops when one of the following two events happens:

- when the midpoint m is the root, or
- when the difference between the two endpoints a and b falls within a threshold, that is, when they become very close to each other. We shall set the threshold to **0.0001** for this exercise. Then the midpoint m is calculated as $(a+b)/2$, and this is the approximated root (answer).

The figure on the next page shows the two endpoints a (–3) and b (0), their midpoint m (–1.5), and the function values at these 3 points: $p(a) = -4$, $p(\text{b}) = 5$, $p(m) = 6.125$.

Since $p(m)$ has the same sign as $p(b)$ (both values are positive), this means that m will replace b in the next iteration.

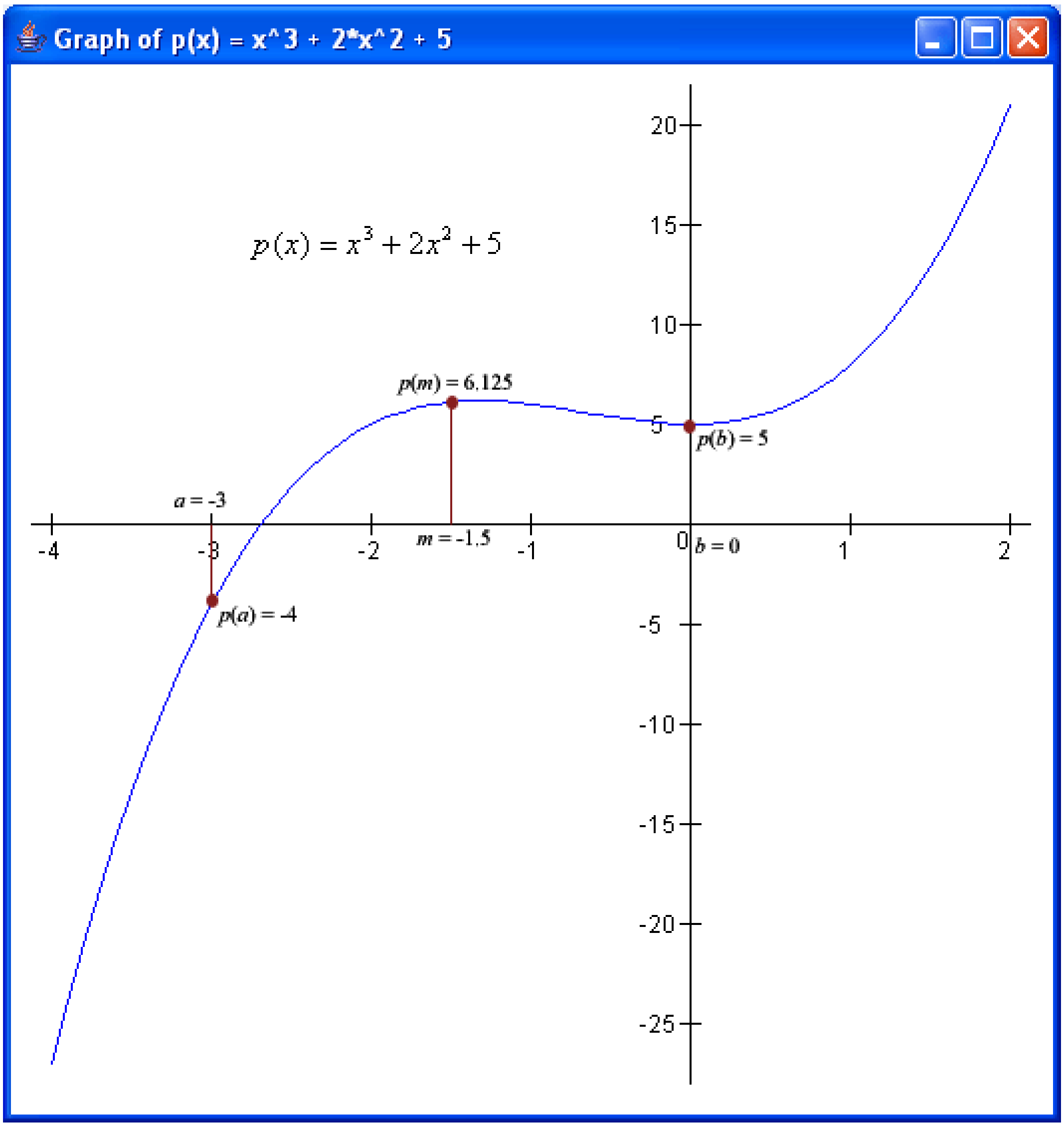

Figure. Graph of $p(x) = x^3 + 2x^2 + 5$

The table on the next page illustrates the iterations in the process. The end-point that is replaced by the mid-point value computed in the previous iteration is highlighted in pink background.

Iteration	endpoint *a*	endpoint *b*	midpoint *m*	Function value *p*(*a*)	Function value *p*(*b*)	Function value *p*(*m*)
1	-3.000000	0.000000	-1.500000	-4.000000	5.000000	6.125000
2	-3.000000	-1.500000	-2.250000	-4.000000	6.125000	3.734375
3	-3.000000	-2.250000	-2.625000	-4.000000	3.734375	0.693359
4	-3.000000	-2.625000	-2.812500	-4.000000	0.693359	-1.427002
5	-2.812500	-2.625000	-2.718750	-1.427002	0.693359	-0.312714
6	-2.718750	-2.625000	-2.671875	-0.312714	0.693359	0.203541
7	-2.718750	-2.671875	-2.695312	-0.312714	0.203541	-0.051243
8	-2.695312	-2.671875	-2.683594	-0.051243	0.203541	0.076980
9	-2.695312	-2.683594	-2.689453	-0.051243	0.076980	0.013077
10	-2.695312	-2.689453	-2.692383	-0.051243	0.013077	-0.019031
11	-2.692383	-2.689453	-2.690918	-0.019031	0.013077	-0.002964
12	-2.690918	-2.689453	-2.690186	-0.002964	0.013077	0.005059
13	-2.690918	-2.690186	-2.690552	-0.002964	0.005059	0.001048
14	-2.690918	-2.690552	-2.690735	-0.002964	0.001048	-0.000958
15	-2.690735	-2.690552	-2.690643	-0.000958	0.001048	0.000045
16	-2.690735	-2.690643	-2.690689	-0.000958	0.000045	-0.000456

Difference between *a* and *b* is < 0.0001

Hence the root of the above polynomial is **–2.690689** (because the difference between *a* and *b* in the last iteration is smaller than the threshold 0.0001), and the function value at that midpoint *m* is **–0.000456**, close enough to zero.

(Some animations on the bisection method can be found on this website:

http://math.fullerton.edu/mathews/a2001/Animations/RootFinding/BisectionMethod/BisectionMethod.html)

Write a program that asks the user to enter the integer coefficients (c_3, c_2, c_1, c_0) for a polynomial of degree 3: $\boldsymbol{c_3x^3 + c_2x^2 + c_1x + c_0}$.

It then asks for the two endpoints, which are real numbers. You may assume that the user enters a continuous function that has a real root. You may use the **double** data types for real numbers.

To simplify matters, you may also assume that <u>the two endpoints the user entered have function values that are opposite in signs</u>.

Your program should have a function/method

double polynomial(double, int, int, int, int)

to compute the polynomial function value.

In the output, real numbers are to be displayed accurate to 6 decimal places (see output in sample runs below).

Sample runs

The sample run below shows the output for the above example. Note that the iterations end when the difference between the two endpoints is less than 0.0001, and the result (root) is the midpoint of these two endpoints.

Only the last 2 lines shown in the sample run below are what your program needs to print. The iterations are shown here for checking purpose only.

```
Enter coefficients (c3,c2,c1,c0) of polynomial: 1 2 0 5
Enter endpoints a and b: -3 0
#1: a = -3.000000; b = 0.000000; m = -1.500000
     p(a) = -4.000000; p(b) = 5.000000; p(m) = 6.125000
#2: a = -3.000000; b = -1.500000; m = -2.250000
     p(a) = -4.000000; p(b) = 6.125000; p(m) = 3.734375
#3: a = -3.000000; b = -2.250000; m = -2.625000
     p(a) = -4.000000; p(b) = 3.734375; p(m) = 0.693359
#4: a = -3.000000; b = -2.625000; m = -2.812500
     p(a) = -4.000000; p(b) = 0.693359; p(m) = -1.427002
#5: a = -2.812500; b = -2.625000; m = -2.718750
     p(a) = -1.427002; p(b) = 0.693359; p(m) = -0.312714
(... omitted for brevity ...)
#15: a = -2.690735; b = -2.690552; m = -2.690643
     p(a) = -0.000958; p(b) = 0.001048; p(m) = 0.000045
#16: a = -2.690735; b = -2.690643; m = -2.690689
     p(a) = -0.000958; p(b) = 0.000045; p(m) = -0.000456
root = -2.690689
p(root) = -0.000456
```

The second sample run below shows how to find the square root of 5. For polynomial where there are more than one real root, only one root needs to be reported.

```
Enter coefficients (c3,c2,c1,c0) of polynomial: 0 1 0 -5
Enter endpoints a and b: 1.0 3.0
#1: a = 1.000000; b = 3.000000; m = 2.000000
     p(a) = -4.000000; p(b) = 4.000000; p(m) = -1.000000
#2: a = 2.000000; b = 3.000000; m = 2.500000
     p(a) = -1.000000; p(b) = 4.000000; p(m) = 1.250000
(... omitted for brevity ...)
#16: a = 2.236023; b = 2.236084; m = 2.236053
     p(a) = -0.000201; p(b) = 0.000072; p(m) = -0.000065
root = 2.236053
p(root) = -0.000065
```

The third sample run below shows how to find the root of the function $2x^2 - 3x$. Since the midpoint of the given endpoints *a* and *b* is 1.5 which is the root of the function, the loop ends after the first iteration.

```
Enter coefficients (c3,c2,c1,c0) of polynomial: 0 2 -3 0
Enter endpoints a and b: 0.5 2.5
#1: a = 0.500000; b = 2.500000; m = 1.500000
     p(a) = -1.000000; p(b) = 5.000000; p(m) = 0.000000
root = 1.500000
p(root) = 0.000000
```