Lab04

Find the zeros of a polynomial

Background: The zeros of a polynomial f(x) are the values of x for which f(x) = 0. In an interval (xlow, xhigh), a monotonic function f(x) has a zero in that interval if either of the following two things are true:

a. f(xlow) is postive and f(xhigh) is negative, or
b. f(xlow) is negative and f(xhigh) is positive.

In the first case, we say the function f(x) is monotonically non-increasing in the interval because in the interval, f(x) never increases in going from xlow to xhigh. Figure 1 is an example of a monotonically non-increasing function in the interval.

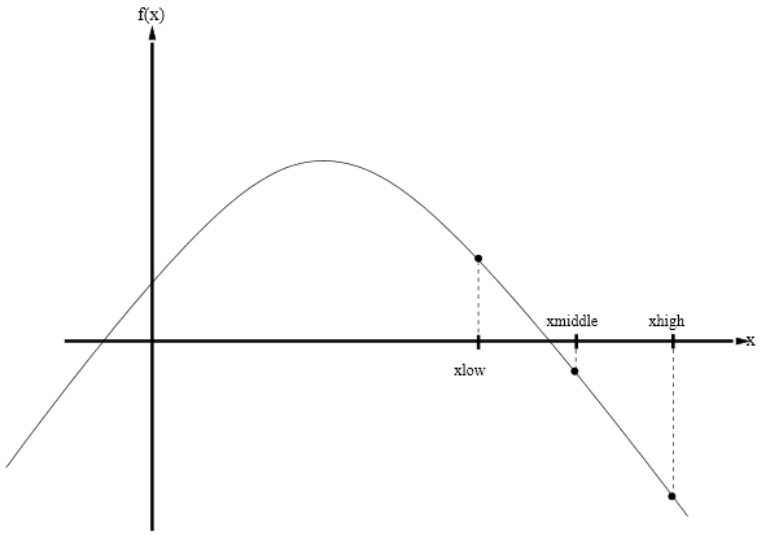

Figure 1

In the second case, we say the function f(x) is monotonically non-decreasing, because in the interval, f(x) never decreases in going from xlow to xhigh.

Your Job: Given an interval (xlow, xhigh), a function f(x) that is monotonic in that interval, and either f(xlow) is positive and f(xhigh) is negative or f(xlow) is negative and f(xhigh) is positive, write an LC-3 assembly language program that uses binary search to find the zero of f(x) in the interval (xlow, xhigh), and store that value of x in x4000.

Input to your program will be found in a table in memory, starting at x4001:

Address	**Content**
x4001	xlow

x4002	xhigh
x4003	degree of the polynomial, say n
x4004 to x4004 + n	polynomial coefficients

All values in the table are 2's complement integers.

For example, if f(x) is the polynomial Ax^3 + Bx^2 + Cx + D, the table will take the form:

Content of memory locations x4000-x4007

Address	**Content**
x4000	output: x-position of zero
x4001	left bound
x4002	right bound
x4003	degree: 3
x4004	A
x4005	B
x4006	C
x4007	D

To evaluate the polynomial, you will need to write a subroutine that computes the value of f(x). This must be written as a subroutine. The starting address of the subroutine should be location x5000. The main program should put the value of x into R0 BEFORE calling the subroutine. The subroutine will take the value in R0 and the coefficients stored in memory and compute the value f(x). The subroutine will place f(x) in R4 before returning to the main program. All other registers (other than R7) should remain unchanged. Your subroutine should work exactly as described. **We should be able to replace your subroutine with one that we have written and your program should still function normally.** The subroutine should be called as few times as possible, part of your grade will be based on this. Please see the binary search section for details.

Sample input and output

Polynomial	**Interval**	**Output**
f(x) = 2x + 6	(-100,100)	-3
f(x) = 4x - 8	(-100,100)	2
f(x) = x^2 - 4x	(2,100)	4
f(x) = x^2 - 4x	(-100,2)	0
f(x) = $4x^2$ - 12x - 16	(2,50)	4
f(x) = x^3 - $15x^2$ + 75x - 117	(-20,20)	3

Binary Search: In our program a user specifies an interval of the function that is monotonic. If f(xlow) is positive and f(xhigh) is negative, then we know that a zero of the polynomial must exist between the two bounds. We can search for the zero by using a **binary search**. In a binary search, we start by dividing the interval into two halves as shown in Figure 1. We can test the polynomial

evaluated at the mid-point of the interval (f(xmiddle)) to see if it is positive, negative or zero. If it is zero, then we are done. If the value is not zero, then we can narrow our search to the interval between xmiddle and the appropriate bound such that the new interval will still contain the zero. In the case of the Figure 1, we can narrow our search to the interval (xlow, xmiddle) because we know that this interval contains the zero. We can discard the interval (xmiddle, xhigh) because we know that it does not contain the zero. This process can be repeated until the zero is found. Question: Why does this algorithm lead to the fewest number of calls (on average) to the f(x) subroutine?

You may assume:

- xlow is always smaller than xhigh.
- f(x) can always be expressed with 16 bits.
- All intervals will contain a zero.
- If f(x) = 0, then x is an integer.
- f(xlow) and f(xhigh) are not zeros.
- x1 + x2 can always be expressed with 16 bits for all values of x1 and x2 in the interval

Hints:

- **Hint 1:** Horner's rule states that Ax^3 + Bx^2 + Cx + D = ((Ax + B)x + C)x + D. How can our subroutine use this to compute f(x)?
- **Hint 2:** You may find it useful to write a multiply subroutine and a divide by 2 subroutine.
- **Hint 3:** When (xhigh + xlow)/2 is not an integer, we will need to round the value to a nearby integer. Is this a problem?

Note: Test your program thoroughly before you submit. Make sure there are no silly mistakes. Please note that there will be NO regrades for this programming lab under any circumstances.

Attention:

1. **Your zip file should contain at least two files:**
   .asm file and report in pdf format.
   As submission format, please refer to the notice on course web page.
2. **Your report should contain at least four parts:**
   purpose, principles, procedure and result.
   Well written will bring you a high score.