CSC 336 MIDTERM TEST 17 October 2019.

This is a **closed-book test**: no books, no notes, no calculators, no phones, no tablets, no computers (of any kind) allowed.

Duration of the test: 50 minutes (6:10 PM to 7:00 PM).

Do **NOT** turn this page over until you are **TOLD** to start.

Answer **ALL** Questions.

Write your answers in the test booklets provided.

Please fill-in **ALL** the information requested on the front cover of **EACH** test booklet that you use.

The test consists of 5 pages, including this one. Make sure you have all 5 pages.

The test consists of 4 questions. **Answer all 4 questions**. The mark for each question is listed at the start of the question.

The test was written with the intention that you would have ample time to complete it. You will be rewarded for concise well-thought-out answers, rather than long rambling ones. **We seek quality rather than quantity**.

Moreover, an answer that contains relevant and correct information as well as irrelevant or incorrect information will be awarded fewer marks than one that contains the same relevant and correct information only.

Write legibly. Unreadable answers are worthless.

1. [5 marks: 1 mark for each answer]

Consider a floating-point number system with parameters $\beta = 10$, $p = 3$, $L = -10$ and $U = +10$ that uses the *round-to-nearest* rounding rule and allows gradual underflow to subnormal numbers as well as underflow to zero. That is, the numbers in the system include zero and nonzero numbers of the form $\pm d_1.d_2d_3 \cdot 10^n$ where $d_i \in \{0, 1, 2, \ldots, 9\}$ for $i = 1, 2, 3$ and $n \in \{-10, -9, -8, \ldots, 10\}$. The normalized floating-point numbers in this system include 0 and the nonzero numbers of the form $\pm d_1.d_2d_3 \cdot 10^n$ with $d_1 \neq 0$. The subnormal numbers have $n = -10$, $d_1 = 0$ and $d_i \neq 0$ for at least one of $i = 2$ or $i = 3$. Like the IEEE floating-point number system, this number system also has the two special numbers +Infty and −Infty, which stand for numbers that are too large in magnitude (either positive or negative, respectively) to represent in this floating-point system. The system also has a NaN, which stands for "not-a-number".

In the floating-point number system described above, what is the result of each of the floating-point arithmetic operations (a)–(e) below? Write your answer as

- a normalized number in this floating-point system, if possible,
- a subnormal number in this floating-point system in the case of gradual underflow,
- zero in the case that the true answer is zero or there is an underflow to zero,
- +Infty or −Infty in the case of overflow,
- NaN if the result of the computation is not any of the above.

(a) $(4.23 \cdot 10^3) + (3.82 \cdot 10^1)$

(b) $(4.08 \cdot 10^5) \times (1.07 \cdot 10^{-2})$

(c) $(5.04 \cdot 10^{-8}) \times (-4.02 \cdot 10^{-4})$

(d) $(-4.01 \cdot 10^3) \times (5.06 \cdot 10^7)$

(e) $(2.05 \cdot 10^{-8}) \times (-4.03 \cdot 10^{-5})$

2. [10 marks: 5 marks for each part]

Jim wrote the MatLab function

```
function [r1,r2] = roots(a,b,c)
   r1 = ( -b + sqrt(b^2 - 4*a*c) ) / (2*a) ;
   r2 = ( -b - sqrt(b^2 - 4*a*c) ) / (2*a) ;
```

to compute the two roots, `r1` and `r2`, of the quadratic $\mathtt{a}x^2 + \mathtt{b}x + \mathtt{c}$.

For $\mathtt{a} = 2$, $\mathtt{b} = 10^8$ and $\mathtt{c} = 4$ his function returned the values

$$\mathtt{r1} = -4.0978 \times 10^{-8} \qquad \text{and} \qquad \mathtt{r2} = -5.0000 \cdot 10^7$$

However, he knew that something was wrong, because he remembered from a high-school math course that the true roots, r_1 and r_2, of the quadratic $\mathtt{a}x^2 + \mathtt{b}x + \mathtt{c}$ satisfy $\mathtt{a} \cdot r_1 \cdot r_2 = \mathtt{c}$, but his computed roots satisfied $\mathtt{a} \cdot \mathtt{r1} \cdot \mathtt{r2} = 4.0978$, but $\mathtt{c} = 4$. So, he knew that at least one of the two roots he calculated must be inaccurate.

Jim checked his function carefully, but he couldn't find anything wrong with it.

(a) Why did Jim's function compute such an inaccurate result?

[Note: although rounding error should play a role in your answer, there should be more to your explanation than just saying that there is rounding error in the computation, since there is rounding error in almost all floating-point computations, but most of them are accurate.]

(b) Advise Jim on how to modify his function so that both computed roots are accurate.

Explain why you believe your modification will produce accurate values for both roots.

3. [5 marks]

Let

$$x = \begin{pmatrix} -1 \\ 3 \\ -2 \end{pmatrix} \qquad \text{and} \qquad A = \begin{pmatrix} -2 & 0 & -4 \\ -2 & 5 & -3 \\ 3 & -1 & 4 \end{pmatrix}$$

Give the value of each of the following norms.

(a) $\|x\|_1$

(b) $\|x\|_2$

(c) $\|x\|_\infty$

(d) $\|A\|_1$

(e) $\|A\|_\infty$

You don't have to try to evaluate numbers like $\sqrt{14}$. Just leave it in your answer as $\sqrt{14}$.

4. [10 marks: 5 marks for each part]

Consider the system of linear equations $Ax = b$, where

$$A = \begin{pmatrix} 2 & -4 & 2 \\ 1 & 1 & -2 \\ 1/2 & 0 & 3/2 \end{pmatrix} \qquad \text{and} \qquad b = \begin{pmatrix} 4 \\ -1 \\ 2 \end{pmatrix}$$

(a) Find the *elementary elimination matrices* M_1 and M_2 and the upper-triangular matrix U such that

i. $M_1 A$ has zeros in its first column below the main diagonal, and

ii. $M_2 M_1 A = U$ has zeros in its first and second columns below the main diagonal (i.e., U is an upper triangular matrix).

Show all your calculations.

(b) Use M_1, M_2 and U to solve $Ax = b$.

Show all your calculations.

Hint: recall that the *elementary elimination matrices* M_1 and M_2 are of the form

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} \qquad \text{and} \qquad M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix}$$

where *the multipliers* m_{21}, m_{31} and m_{32} are real numbers. Finding M_1 and M_2 in part (a) essentially amounts to finding values for the multipliers m_{21}, m_{31} and m_{32}.