Please write your family and given names and **underline** your family name on the front page of your paper.

All questions **must** be written in latex, and compiled to a single pdf. All code and output of Q1 and Q3 should be *embedded* into latex. Code and output should be embedded with *fixed-width fonts*, e.g. Courier. Font size of all fonts must be 12.
What to submit:
(1) The single pdf file a1.pdf (with embedded code and output).
(2) The latex source file of A1.
(3) Any source code you wrote (for computation, etc).
Thus, the code will be available within latex/pdf, *as well as* separately.

**1.**

(a) [*10 points*] Find the condition number of $f(x) = (a + x)^{1/4} - a^{1/4}$, for $x > 0$, $a > 0$ and study whether there are ranges of $x \in \mathbb{R}^+$ for which the computation of $f$ is ill-conditioned. (You may need to use de l' Hospital's rule.)

(b) [*10 points*] Consider the (numerical) stability of the computation of the expression $(a + x)^{1/4} - a^{1/4}$, for $x > 0$, $a > 0$, when $x$ is close to 0. Explain what problems the computation of the expression may give rise to. Propose a mathematically equivalent expression that is likely to be more stable for $x$ close to 0, and explain.

(c) [*10 points*] Set $a = 1$. Write a MATLAB or equivalent script that goes through the values of $x$ in $\{10^{-20}, 10^{-19}, \ldots, 10^{-1}, 1, 10, \cdots, 10^{19}, 10^{20}\}$, and computes and outputs $x$, the respective values of $f$ using the original expression, as well as your proposed (more stable) expression, the respective condition numbers (computed using the values of $f$ with the original expression and the values of $f$ with the proposed expression), and the relative error between the $f$ values computed using the original and proposed expressions. Comment on the results. Use an appropriate format for the results, e.g. `fprintf('%9.2e %12.5e %12.5e %12.5e %12.5e %10.2e0, ...`

**2.** Consider the integrals

$$y_n = \int_0^1 t^n e^{-t} dt, \quad n = 0, 1, 2, \cdots$$

(a) [*10 points*] Derive a recurrence relation for $y_n$ relating $y_n$ to $y_{n-1}$. (You may have to use integration by parts.) Rearrange the formula so that you have a recurrence relation for $y_{n-1}$ relating $y_{n-1}$ to $y_n$. Name the first recurrence formula (A) and the second one (B).

(b) [*10 points*] With repeated applications of (A), give a formula that gives $y_n$ as a function of $y_0$, i.e. $y_n = f_n(y_0)$. With repeated applications of (B), give a formula that gives $y_n$ as a function of $y_m$, for $m > n$, i.e. $y_n = g_{n,m}(y_m)$.

(c) [*10 points*] Find the condition number of the functions

$$f_n(y_0) \text{ and } g_{n,m}(y_m) \text{ for } m > n.$$

(Note: Function $f_n$ has $y_0$ as the variable, and function $g_{n,m}$ has $y_m$ as the variable.)
Taking into account the condition numbers of the above two functions formulate a stable method for computing $y_0, y_1, \cdots, y_N$, where $N \geq 1$ is given.

(d) [*10 points*] Write and run a MATLAB or equivalent program that computes and outputs $y_0, y_1, \cdots, y_N$, starting with $y_0$ and using recursion (A). Explain what happens! (A reasonable $N$ to stop is $N = 20$.)

(e) [*15 points*] Write a MATLAB or equivalent program that sets $y_{N+K}$ to some appropriate value (which may be approximate), and computes and outputs $(N + K - 1, y_{N+K-1}), (N + K - 3, y_{N+K-2}), \cdots, (N, y_N)$, starting with $y_{N+K}$ and using recursion (B). At the end of the recursion, when the value of $y_N$ is output, also output the "exact" value and the (absolute) error in the computed $y_N$. Run the program for $K = 3, \cdots, 9$ and $N = 20$ (7 cases). Note that the code should have a nested loop (one loop for $K$ and one for the recursion). Explain what happens! Assume the "exact" value is `q = 0.018350467697256206326;` Comment on how one should compute $y_{20}$ to machine precision.

Notes: You should not use any symbolic environment. No integral calculations (other than the recurrences given). Use (the standard) double precision. You should not change the precision. Use an appropriate format, e.g. `fprintf('%3d %20.16f\n', i-1, y(i));` and `fprintf('%3d %20.16f %20.16f %10.6e\n', 20, y(21), q, q-y(21));`.

**3.** [*15 points*] Assume that $A$ and $B$ are given dense $n \times n$ matrices, $B$ is non-singular, $\mathbb{I}$ is the identity matrix of order $n$, and $b$ a given $n \times 1$ vector, for some $n$ large. Explain how you would efficiently compute $z = B^{-1}(2A + \mathbb{I})(B^{-1} + A)b$. Give, in terms of $n$, approximate operation counts for all computations you propose.

*Note*: The computations that you will propose may include LU factorization, back-and-forward substitutions, matrix-vector products, matrix-matrix products, matrix inverse calculation, addition of vectors or matrices, and other similar computations. However, you are **not** obliged to use **all** these types of computations. For each computation you propose, you should give operation counts (indicating the highest power of $n$, including the coefficient), and make sure that the total number of operation counts is as little as possible. You do **not** need to describe algorithms for these computations. Also note that, while $B$ is given, $B^{-1}$ is not given.