The final will consist of two parts.
Part one is recorded below.
Part two will be released later in the week.

Part 1.
Write python code for the barrier algorithm as outlined in the course text.

More precisely, create a single .py file containing python functions for each of the following algorithms:

- backtracking line search
- KKT system builder and solver
- Newton's method for equality constrained problems using previous two functions.
  (You can use feasible start Newton's method.)
- Barrier method whose centering step uses your Newton's method code.

Refer to text for pseudocode for the algorithms.

Note: You may only import the libraries jax.numpy and jax for part 1.

Using other libraries or functions which trivialize the code/problem will be marked down heavily.
(E.g., don't just import and use cvxpy.)

More precisely, the python barrier method function must take in the following arguments

- objective function f_0
- initial point $x^{(0)}$
- multiplier μ
- tolerance ϵ
- backtracking parameters α, β
- matrix A and vector b encoding equality constraint $Ax = b$
- list of inequality constraint functions $f_1, \ldots, f_m$.

The output should be a feasible point $x^\star$ which minimizes f_0 up to tolerance ϵ, i.e., $f(x^\star) - p^\star < \epsilon$.

Additionals:

- Your barrier method function should be able to handle the cases:
  - no inequality constraints
  - no equality constraint
  - neither inequality nor equality constraints
- You will submit a single .py file for this part of the final.