```
%%%%%%%%%%%%%%% Main Entrance %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%% By Max Yi Ren and Emrah Bayrak %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Instruction: Please read through the code and fill in blanks
% (marked by ***). Note that you need to do so for every involved
% function, i.e., m files.

%% Optional overhead
clear; % Clear the workspace
close all; % Close all windows
clc;

%% Optimization settings
% Here we specify the objective function by giving the function handle to a
% variable, for example:
f = @(x)objective(x); % replace with your objective function
% In the same way, we also provide the gradient of the
% objective:
df = @(x)objectiveg(x); % replace accordingly

g = @(x)constraint(x);
dg = @(x)constraintg(x);

% Note that explicit gradient and Hessian information is only optional.
% However, providing these information to the search algorithm will save
% computational cost from finite difference calculations for them.

% Specify QP solution algorithm
% When set to 'matlabqp' MATLAB's QP solver is used.
% When set to 'myqp' your own QP solver is used.

opt.alg = 'myqp'; % 'myqp' or 'matlabqp'

% Turn on or off line search. You could turn on line search once other
% parts of the program are debugged.
opt.linesearch = true; % false or true

% Set the tolerance to be used as a termination criterion:
opt.eps = 1e-3;

% Set the initial guess: (column vector, i.e. x0 = [x1; x2] )
x0 = [1; 1];
disp(g(x0));
% Feasibility check for the initial point.
if max(g(x0)>0)
    errordlg('Infeasible intial point! You need to start from a feasible one!');
    return
end
%% Run optimization
% Run your implementation of SQP algorithm. See mysqp.m
solution = mysqp(f, df, g, dg, x0, opt);

%% Report
report(solution,f,g);
```