The instructor is aware of the possibilities that exist for obtaining homework solutions outside of one's own abilities. Therefore, read the following rules carefully!

- You can collaborate or work with anyone. However, the work submitted must be your understanding. If your works contain seems quite similarity, you can **share the total points**.
- If your work involves significant help from reference books or papers, add a note in your paper to indicate this. The murkiness about "using online tutorials" is resolved below:
  - Do not submit your homework problems to someone or some agency who will do the exam for you. These people do everyone except themselves a disservice. They harm you because you won't learn and therefore won't earn (neither money nor the joy of knowing something). They mock the education process. They benefit by collecting cash because you may not know something. Don't let this be a pattern for your life. **A grade is a grade and over time will mean less and less, but knowledge acquired pays dividends for as long as you retain that knowledge.**
- **The instructor reserves the right to ask the student to explain the answers for any or all problems on the homework.** If the student is unable to provide a satisfactory answer, then it will be assumed that the work was not done in an earnest manner and as such the problem in question will receive no credit.
- Show all your work. Correct answers without sufficient explanation might **not** get full credit.
- The assignment consists of 7 questions for a total of 100 marks.
- Submit your assignment **electronically** via https://odtuclass.metu.edu.tr/.
- When you scan your hand-writing part, please make sure that one can easily read the homework when printed. Do not submit photos of handwriting homework.
- The report of MATLAB part should be written via **LaTeX**.
- Please, do not forget send **.m files**, **.tex file** of MATLAB part.
- Combine the hand-writing part and the report of MATLAB part in **a single pdf**.
- Please, do not forget write your name and surname to your documents.
- Together with your MATLAB **.m files**, make a compressed file such as ZIP (never use RAR!), of all your work (including your PDF report (.pdf and .tex files)). Upload only the compressed file having a name as **your name, surname, and homework number**, for instance, **mkutuk_hmw1**.
- The report for a programming part should consist of
  - Brief description of the problem;

  - Results such as data, graphs etc. Try to avoid attaching large set of data unless it is really necessary;
  - Discussion, comments, explanation and conclusion on your numerical observations. This part is equally important as your computer codes and the data you collect, and it helps with understanding concepts, algorithms, or other relevant issues not discussed during lectures.
  - Please, do not insert your MATLAB **.m files** into the report if not necessary.

- Every .m file should be documented, such as:

```matlab

% Author: (Your Name)
%
% Description:
% (Give a brief description of what your program does.)
%
% Input:
% (State what the program inputs are.)
%
% Output:
% (State what the program outputs are.)
%
% Usage:
% (Give an example of how to use your program.)
```

- **Late submission is not allowed!**

- **Please write the exact wording of the Pledge, following by your signature and date, in the beginning of your paper. Otherwise, your homework will not be evaluated.**

  ”*I confirm that I have read the instructions carefully and fully understood my responsibilities. I hereby declare that I have neither given nor received aid in this homework nor I have concealed any violation of the University Honor code.*”

1. (8 *pts.*) A company owns 800 hectares (ha) of land to build different types of houses based on the following estimates:

Type	**Profit**	**Cost**	**Water Usage**	**Required Area**
I	200	145	20 l per day	1 ha
II	240	165	27 l per day	1.5 ha
III	300	215	32 l per day	2 ha

In addition, the company has to obey the following rules:

- At least half of the houses have to be Type-I.
- Water usage cannot exceed 8500 l per day.
- Recreational areas require 15 % of the total area.
- For every 15 houses, there has to be at least one recreational areas, which requires 0.5 ha of area. In addition, building cost and water usage per day are 125 and 25 l, respectively.

Formulate a linear programming model that will maximize the profit.

2. (16 *pts.*) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function.

   a) (4 *pts.*) Prove that if the function f is convex, then

   $$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

   Make a comment on the geometric meaning of the above result.

   b) (4 *pts.*) Prove that $f \in \mathcal{C}^2(\mathbb{R}^n)$ is convex if the hessian of f is positive semi–definite for all $x \in \mathbb{R}^n$.

   c) (4 *pts.*) Show that the set $C_\alpha = \{x \in \mathbb{R}^n \ : \ f(x) \leq \alpha \ \}$ is convex if f is convex.

   d) (4 *pts.*) Let the function $g(x)$ be given by

   $$g(x) = \big(c^T x + d\big) f\Big(\frac{Ax + b}{c^T x + d}\Big),$$

   where $dom(g) = \{x \in \mathbb{R}^n : c^T x + d > 0, \ \frac{Ax+b}{c^T x+d} \in dom(f)\}$ and f is a convex function. Then, show that $g(x)$ is also convex.

3. (12 *pts.*) A quadratic function $f(x)$ is defined as follows

$$f(x) = c^T x + \frac{1}{2} x^T H x,$$

where

$$c = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \qquad H = \begin{bmatrix} 4 & 4 & 4 & 3 \\ 4 & 7 & 3 & 3 \\ 4 & 3 & 5 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}.$$

   a) (4 *pts.*) Determine whether a stationary point x^* exists or not.

b) (4 *pts.*) If exists, compute the stationary point x^*. Is x^* a minimizer of f? Justify your answer!

c) (4 *pts.*) If x^* is a minimizer of $f(x)$, is x^* unique? Explain!

4. (14 *pts.*) Consider the following nonlinear system

$$x^2 + y^2 - 2 = 0, \quad (1a)$$
$$x - y = 0. \quad (1b)$$

a) (2 *pts.*) Find the roots of the systems.

b) (3 *pts.*) Write one iteration of Newton's method with the initial guess (x_0, y_0) to find the zeros of the system.

c) (4 *pts.*) Let (x_0, y_0) be the initial guess of the Newton system. Show that the iteration converges to $(1,1)^T$ if $x_0 + y_0$ is positive and converges to $(-1,-1)^T$ if $x_0 + y_0$ is negative.

d) (5 *pts.*) Verify that convergence of the Newton iteration obtained for the system (1) exhibits quadratic convergence.

5. (20 *pts.*) Consider the following objective function

$$f(x_1, x_2) = \frac{x_1^2}{2} + x_1 \cos x_2.$$

a) (3 *pts.*) Find the gradient and Hessian of f.

b) (5 *pts.*) Find all minima of f.

c) (3 *pts.*) Confirm that the search direction $\mathbf{p}_0 = (-1, 0)^T$ is a descent direction at $\mathbf{x}_0 = (0, \pi/4)$.

d) (4 *pts.*) For $\mathbf{p}_k = (-1, 0)^T$ and $\mathbf{x}_k = (0, \pi/4)$, what are the admissible values for the step length α_k for the Wolfe conditions, which are

$$\begin{aligned} f(\mathbf{x} + \alpha_k \mathbf{p}_k) &\leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{p}_k, \\ \nabla f(\mathbf{x} + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k &\geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{p}_k \end{aligned}$$

for which $c_1 = 0.1$ and $c_2 = 0.8$ are chosen?

e) (5 *pts.*) Perform one step of the exact line search for the search direction $\mathbf{p}_0 = (-1, 0)$ at $\mathbf{x}_0 = (0, \pi/4)$.

6. (15 *pts.*) In this exercise, you will write a Matlab routine of Newton's method `newton.m` to find the roots of a given function $f(x)$. Read the following instructions:

- Write the Newton method code of the form:

```
[x,hist, hist_err, iters] = Newton(f,x_0,tol,maxit);
Input:
f     : a user supplied function
x_0   : initial guess
tol   : a positive real number (the stopping tolerance)
maxit: a positive integer specifying the max number
```

```
of iterations allowed.
Output:
x        : approximate solution to f(x) = 0
hist     : an array (a vector) of the values of x_k
hist_err : an array (a vector) of the error, i.e.,  x^* - x_k
iter     : the number of iterations taken
```

- Call the function `f` as `[f, df] = f(x)` which returns the values of the function f as its derivative $\frac{df}{dx}$.
- Stopping criteria is applied as

$$|f(x_k)| < tol,$$

  where tol is a user specified stopping tolerance and $f(x_*) = 0$.
- Run your code for different values of `tol = tol_1, tol_2, tol_3`, initial guess $x_0 = 1.8$, $maxit = 1000$, and functions $f(x) = (x^2 + 1)(x - 1)$ and $g(x) = (x - 1)^2$.
- Display your results as the following:

```
Tol   | iteration     |     x value    | error
----- |--------- ----|------------------------
1e-3  |   iters_1     |      x_1       |  err_1
1e-6  |   iters_2     |      x_2       |  err_2
1e-9  |   iters_3     |      x_3       |  err_3
```

7. (15 *pts.*) In this exercise, you will write a Matlab routine to implement **Newton method with Armijo condition** in the form of

```
function [X,Grad,it] =
Newton_armijo(fhandle,x0,tol,maxit,alpha0,c,mu,amax)
```

taking the following input arguments:

- objective function `fhandle`, initial guess `x0`, tolerance value for the termination condition $\|\nabla f(x_k)\| <$ `tol`, maximum number of iteration `maxit`, initial step–length `alpha0`, Armijo constant `c`, backtracking parameter `mu`, and maximum number of iterations `amax` for Armijo iteration.

and return

- a matrix `X = [x0; x1; x2; ...]` containing the whole iterations, a matrix `Grad` containing $\|\nabla f(x_k)\|$, and the number of iterations `ite`.

Newton Method with Armijo Condition

Input: Given the objective function $f(x)$, initial guess x_0, tolerance number `tol`, maximum number of iteration `maxit`, Armijo constant c, backtracking constant μ, maximum number of iteration for Armijo `amax`, and initial guess for step–length α_0.

Set $k := 0$.

while $\|\nabla f(x_k)\| \geq$ `tol` and $k \leq$ `maxit` **do**

  Compute the search direction $p_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$.

  Compute α_k by using backtracking approach with Armijo condition:

  Set $j := 0$.

  while $f(x_k + \alpha_j p_k) \geq f(x_k) + c\alpha_j \nabla f(x_k)^T p_k$ and $j \leq$ `amax` **do**

    $\alpha := \alpha * \mu$.

    Set $j := j + 1$.

  end while

  Update the solution $x_{k+1} = x_k + \alpha_k p_k$.

  Set $k := k + 1$.

end while

- Take the Rosenbrock banana function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

  as a benchmark function. Write a file `Rosenbrock.m` for the Rosenbrock banana function `[f,df,Hess] = f(x)`, which returns the values of the function f, gradient $\nabla f(x)$, and Hessian $\nabla^2 f(x)$, respectively.
- Run your code for different values of `tol` $= 10^{-3}, 10^{-6}, 10^{-9}$, initial guesses $x_0 = (-0.5, 1)^T$ and $x_0 = (1.1, 1.1)^T$, `maxit` $= 10000$, `alpha0` $= 1$,`c` $= 1e-4$, `mu` $= 0.5$, and `amax` $= 100$.
- Display your results for each initial guesses as the following:

```
Tol    | iteration     |     x value    | Norm_Gradient
-----  |--------- ----|------------------------
1e-3   |    iters_1    |      x_1       |  grad_1
1e-6   |    iters_2    |      x_2       |  grad_2
1e-9   |    iters_3    |      x_3       |  grad_3
```

  We note that the norm of gradient is given for the last step.
- Plot the norm of the gradient for each tolerance in the same figure.
- In this exercise, you need to write also a function

```
function [alpha] = armijo(fhandle, x, p, alpha0, c, mu, amax)
```

  that returns the step–length that satisfies Armijo condition. As input arguments, the function accepts

  – a function handle `fhandle`, current iterate `x`, descent direction `p`, initial step–length `alpha0`, Armijo constant `c` backtracking parameter `mu`, and maximum number of iterations `amax`.