Assignment 2

Name: Mohd Darish Khan

Roll: 2101MC29

Question 1:

Code :

```
% given aray
a = [1, 4, -9, 7, 9, 4];
% made matrix with two same rows
b = [a; a];

% implementing bubble sort on first row
for i = 1:1:6
    for j = i+1:1:6
        if b(1, j)>b(1,i)
            temp = b(1, i);
            b(1, i) = b(1, j);
            b(1, j) = temp;
        end
    end
end

% displaying the matrix
disp("Final Matrix : ");
disp(b);
```

In function form :

```
function B = sorted_ans(a)
    % made matrix with two same rows
    B = [a; a];

    % implementing bubble sort on first row
    for i = 1:1:length(a)
        for j = i+1:1:length(a)
            if B(1, j)>B(1,i)
                temp = B(1, i);
                B(1, i) = B(1, j);
                B(1, j) = temp;
            end
        end
    end
end
```

Output :

```
Final Matrix :
     9     7     4     4     1    -9
     1     4    -9     7     9     4
```

Question 2 :

Code :

```
% Define the functions F and G
F = @(x, y) x.^2 + y.^2;
G = @(x, y) x .* cos(y);

% Compute the gradients of F and G
grad_F = @(x, y) [2*x, 2*y];
grad_G = @(x, y) [cos(y), -x .* sin(y)];

points = [0, 0; 0, 0.5; 1, 3];

% Direction vector and its unit vector
direction = [1, -2];
unit_direction = direction / norm(direction);

for i = 1:size(points, 1)
    x_val = points(i, 1);
    y_val = points(i, 2);

    F_val = F(x_val, y_val);
    G_val = G(x_val, y_val);

    if F_val > G_val
        grad_H = grad_F(x_val, y_val);
    elseif G_val > F_val
        grad_H = grad_G(x_val, y_val);
    else
        grad_F_val = grad_F(x_val, y_val);
        grad_G_val = grad_G(x_val, y_val);
        if norm(grad_F_val) >= norm(grad_G_val)
            grad_H = grad_F_val;
        else
            grad_H = grad_G_val;
        end
    end

    direc_derivative = dot(grad_H, unit_direction);

    % Output the result
    fprintf('The directional derivative of h at (%.1f, %.1f) in the 
direction of (1, -2) is %.4f\n', x_val, y_val, direc_derivative);
end
```

Output :

```
The directional derivative of h at (0.0, 0.0) in the direction of
(1, -2) is 0.4472
The directional derivative of h at (0.0, 0.5) in the direction of
(1, -2) is -0.8944
The directional derivative of h at (1.0, 3.0) in the direction of
(1, -2) is -4.4721
```

Question 3 :

Code :

```
warning('off', 'all');
% Part a
xRange = linspace(-2, 2, 500);
tRange = linspace(0, 4, 500);
[X, T] = meshgrid(xRange, tRange);
Z = X.^2;

figure;
hold on;

% Masking values below the function
T(T < Z) = NaN;

% Plot the epigraph
mesh(X, T, Z, 'FaceColor', [0.7 0.8 0.2], 'EdgeColor', 'none');
contour3(X, T, Z, 20, 'LineWidth', 1);

xlabel('x');
ylabel('t');
title('Epigraph of f(x) = x^2');
grid on;
hold off;

% Display convexity of the function
disp('The function f(x) = x^2 is convex as its Epigraph is
convex.');


% Part b
syms x y real;
f2 = exp(x) + y^2;
hessianF2 = computeHessian(f2, x, y);

% Check eigenvalues for non-negativity
convexCheckF2 = all(arrayfun(@(ev) isAlways(ev >= 0), hessianF2));

if convexCheckF2
    disp('The function f(x, y) = exp(x) + y^2 is convex.');
else
    disp('The function f(x, y) = exp(x) + y^2 is not convex.');
end

% Part c
f3 = sin(x) + cos(y);
hessianF3 = computeHessian(f3, x, y);
```

```
% Check eigenvalues for non-negativity
convexCheckF3 = all(arrayfun(@(ev) isAlways(ev >= 0), hessianF3));

if convexCheckF3
    disp('The function f(x, y) = sin(x) + cos(y) is convex.');
else
    disp('The function f(x, y) = sin(x) + cos(y) is not convex.');
end


% Part d
f4 = x^3 + y^3;
hessianF4 = computeHessian(f4, x, y);

% Check eigenvalues for non-negativity
convexCheckF4 = all(arrayfun(@(ev) isAlways(ev >= 0), hessianF4));

if convexCheckF4
    disp('The function f(x, y) = x^3 + y^3 is convex.');
else
    disp('The function f(x, y) = x^3 + y^3 is not convex.');
end


% Part e
f5 = x^3 + y^3;
hessianF5 = computeHessian(f5, x, y);

% Check convexity in the domain [4, 7] x [4, 7]
xRangeMin = 4; xRangeMax = 7;
yRangeMin = 4; yRangeMax = 7;
isConvexInRegion = true;

for i = xRangeMin:xRangeMax
    for j = yRangeMin:yRangeMax
        eigenValuesAtPoint = subs(hessianF5, {x, y}, {i, j});
        if any(eigenValuesAtPoint < 0)
            isConvexInRegion = false;
            break;
        end
    end
    if ~isConvexInRegion
        break;
    end
end

if isConvexInRegion
    disp('The function f(x, y) = x^3 + y^3 is convex in the domain
[4, 7] × [4, 7].');
else
```

```
    disp('The function f(x, y) = x^3 + y^3 is not convex in the
domain [4, 7] × [4, 7].');
end


function eigVals = computeHessian(func, xVar, yVar)
    gradX = diff(func, xVar);
    gradY = diff(func, yVar);

    secondGradXX = diff(gradX, xVar);
    secondGradYY = diff(gradY, yVar);
    mixedGradXY = diff(gradX, yVar);

    H = [secondGradXX, mixedGradXY; mixedGradXY, secondGradYY];

    eigVals = simplify(eig(H));
end
```

Output :

```
The function f(x) = x^2 is convex as its Epigraph is convex.
The function f(x, y) = exp(x) + y^2 is convex.
The function f(x, y) = sin(x) + cos(y) is not convex.
The function f(x, y) = x^3 + y^3 is not convex.
The function f(x, y) = x^3 + y^3 is convex in the domain [4, 7] ×
[4, 7].
```