Contents

QUESTION 1 (LINEAR ALGEBRA)

```
%Initialization
clear all
% Define the 16 statements
statements = {
    'Each superset of a linearly dependent set is linearly dependent.'
    'Each subset of a linearly independent set is linearly independent.'
    'Union of any two linearly dependent sets is linearly dependent.'
    'Intersection of any two linearly independent sets is linearly independent.'
    'Each subset of a linearly dependent set is linearly dependent.'
    'Each superset of a linearly independent set is linearly independent.'
    'Union of any two linearly independent sets is linearly independent.'
    'Intersection of any two linearly dependent sets is linearly dependent.'
    'If span(A) ∩ span(B) = {0}, then A ∪ B is linearly independent.'
    'If v1, . . . , vn are linearly independent, then v1, v2 − v1, . . . , vn − v1 are linearly independent.'
    'If v1, . . . , vn span V, then v1, v2 −v1, . . . , vn −v1 span V.'
    'If v1, . . . , vn are linearly independent, then v1, v2 − v1, . . . , vn − v1 are linearly dependent.'
    'span(span(A)) = span(A).'
    'If A ⊆ B, then span(A) ⊆ span(B).'
    'span(A ∩ B) ⊆ span(A) ∩ span(B).'
    'span(A) ∩span(B) ⊆ span(A ∩ B).'
};

for i=1:5
%Title
fprintf('\nQ1V%d\n ',i);

    % Randomly select 4 statements
    selected_indices = randperm(16, 4);

    % Display the selected statements
    disp('Which of the following statements are correct?')
    options = char('a' + (1:numel(selected_indices)) - 1);
    for i = 1:numel(selected_indices)
        disp([char(options(i)) ') ' statements{selected_indices(i)}])
    end
    disp(['e) None of the above'])

    % Generate the answer key and explanations for the selected statements
    answer_key = [1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 0]; % Correctness of each statement
    explanations = {
        'Let [p,q] be linearly dependent set so ap+bq = 0 if [p,q,r] is superset we can write ap+bq+(0)r = 0 and atleast one of the a,b is not zero, so the
        'Use the notion of linear independence'
        'Use the notion of linear dependence'
        'Use the notion of linear independence'
        'Take any linearly independent set and add the zero vector to it'
        'Take any linearly independent set and add the zero vector to it'
        'consider Set A: {v1, v2} where v1 = [1, 0] and v2 = [0, 1]Set B: {v3, v4} where v3 = [2, 0] and v4 = [0, 2]'
        'consider Set A: {v1, v2} where v1 = [1, 0] and v2 = [0, 1]Set B: {v2, v3} where v2 = [0, 1] and v3 = [1, 1]'
        'Set A: {v1} where v1 = [1, 0] Set B: {v2} where v2 = [0, 1]'
        'Trivial'
        'It is enough to show that the given set of vectors spans the vectors v1, v2, ..., vn.'
        'Trivial'
        'Follows from the fact that L(s) is the subspace'
        'Trivial'
        'Take an element form L(A ∩ B), write it as linear combination of elements of A ∩ B and observe'
        'Take two different non-zero vectors such that linear span is the same space'

    };

    % Create options strings
    options = char('a' + (1:numel(selected_indices)) - 1);
    options = [options 'e'];



    % Check the correctness of the answer
    correct_options = options(answer_key(selected_indices) == 1);
    incorrect_options = options(answer_key(selected_indices) == 0);

    % Display the answer

    if isempty(correct_options)
        disp('answer is e) None of the above')
    else
        disp(['The correct options are: ' char(correct_options)])

    end
```

```
        % Explanation of the answer
        disp('Explanation:')


            for i = 1:numel(selected_indices)
                index = selected_indices(i);
                %statement = statements{index};
                %answer = answer_key(index);
                explanation = explanations{index};
                %([char(options(i)) ') ' statement])
                %disp(['   Correct answer: ' num2str(answer)])
                disp([char(options(i)) ') ' explanation])
            end

    end
```

```
Q1V1
 Which of the following statements are correct?
a) span(A) ∩span(B) ⊆ span(A ∩ B).
b) If A ⊆ B, then span(A) ⊆ span(B).
c) If v1, . . . , vn are linearly independent, then v1, v2 − v1, . . . , vn − v1 are linearly independent.
d) Union of any two linearly independent sets is linearly independent.
e) None of the above
The correct options are: bc
Explanation:
a) Take two different non-zero vectors such that linear span is the same space
b) Trivial
c) Trivial
d) consider Set A: {v1, v2} where v1 = [1, 0] and v2 = [0, 1]Set B: {v3, v4} where v3 = [2, 0] and v4 = [0, 2]

Q1V2
 Which of the following statements are correct?
a) If v1, . . . , vn are linearly independent, then v1, v2 − v1, . . . , vn − v1 are linearly independent.
b) If v1, . . . , vn are linearly independent, then v1, v2 − v1, . . . , vn − v1 are linearly dependent.
c) Each subset of a linearly dependent set is linearly dependent.
d) span(span(A)) = span(A).
e) None of the above
The correct options are: ad
Explanation:
a) Trivial
b) Trivial
c) Take any linearly independent set and add the zero vector to it
d) Follows from the fact that L(s) is the subspace

Q1V3
 Which of the following statements are correct?
a) span(A ∩ B) ⊆ span(A) ∩ span(B).
b) Union of any two linearly dependent sets is linearly dependent.
c) Each superset of a linearly dependent set is linearly dependent.
d) Intersection of any two linearly dependent sets is linearly dependent.
e) None of the above
The correct options are: abc
Explanation:
a) Take an element form L(A ∩ B), write it as linear combination of elements of A ∩ B and observe
b) Use the notion of linear dependence
c) Let [p,q] be linearly dependent set so ap+bq = 0 if [p,q,r] is superset we can write ap+bq+(0)r = 0 and atleast one of the a,b is not zero, so the superset
d) consider Set A: {v1, v2} where v1 = [1, 0] and v2 = [0, 1]Set B: {v2, v3} where v2 = [0, 1] and v3 = [1, 1]

Q1V4
 Which of the following statements are correct?
a) Union of any two linearly independent sets is linearly independent.
b) Each superset of a linearly dependent set is linearly dependent.
c) Intersection of any two linearly dependent sets is linearly dependent.
d) If v1, . . . , vn span V, then v1, v2 −v1, . . . , vn −v1 span V.
e) None of the above
The correct options are: bd
Explanation:
a) consider Set A: {v1, v2} where v1 = [1, 0] and v2 = [0, 1]Set B: {v3, v4} where v3 = [2, 0] and v4 = [0, 2]
b) Let [p,q] be linearly dependent set so ap+bq = 0 if [p,q,r] is superset we can write ap+bq+(0)r = 0 and atleast one of the a,b is not zero, so the superset
c) consider Set A: {v1, v2} where v1 = [1, 0] and v2 = [0, 1]Set B: {v2, v3} where v2 = [0, 1] and v3 = [1, 1]
d) It is enough to show that the given set of vectors spans the vectors v1, v2, ..., vn.

Q1V5
 Which of the following statements are correct?
a) If v1, . . . , vn are linearly independent, then v1, v2 − v1, . . . , vn − v1 are linearly independent.
b) If v1, . . . , vn span V, then v1, v2 −v1, . . . , vn −v1 span V.
c) Each superset of a linearly independent set is linearly independent.
d) If span(A) ∩ span(B) = {0}, then A ∪ B is linearly independent.
e) None of the above
The correct options are: abd
Explanation:
a) Trivial
b) It is enough to show that the given set of vectors spans the vectors v1, v2, ..., vn.
c) Take any linearly independent set and add the zero vector to it
d) Set A: {v1} where v1 = [1, 0] Set B: {v2} where v2 = [0, 1]
```

QUESTION 2 (LINEAR ALGEBRA)

```
    clear all
```

```
for i=1:5
   %Title
   fprintf('\nQ2V%d\n ',i);

   % Define matrices P1 to P6 as a cell array
   k = randi([1,10]);
   P = cell(1, 6);
   P{1} = [k, 0, 0; 0, k, 0; 0, 0, k];
   P{2} = [k, 0, 0; 0, 0, k; 0, k, 0];
   P{3} = [0, k, 0; k, 0, 0; 0, 0, k];
   P{4} = [0, k, 0; 0, 0, k; k, 0, 0];
   P{5} = [0, 0, k; k, 0, 0; 0, k, 0];
   P{6} = [0, 0, k; 0, k, 0; k, 0, 0];

   % Define matrix Q
   % Generate a symmetric matrix of order 3 with elements in range [-100, 100]
   A = randi([-100, 100], 3);  % Generate random integers in the range [-100, 100]

   % Make the matrix symmetric
   Q = (A + A')/2;


   % Compute X
   X = zeros(3);

   for k = 1:6
       X = X + P{k} * Q * P{k}';
   end

   % Calculate the correct answer for the sum of diagonal elements
   tr = trace(Q);
   correctAnswer = k^2 * tr * 6;

   % Generate random permutations of array indices
   indices = randperm(4);

   % Assign jumbled and unique values to variables
   options = [k^2 * tr *6, k^2 * tr * 3, k^2 * tr * 12, k^2 * tr];
   % Generate random permutation of array indices
   permutedIndices = randperm(numel(options));

   % Create a new array with jumbled elements
   jumbledOptions = options(permutedIndices);

   % Determine the correct option index
   correctOptionIndex = find(jumbledOptions == correctAnswer);

   % Display the matrices and the question
   disp('Let ')
   disp('P1:');
   disp(P{1});
   disp('P2:');
   disp(P{2});
   disp('P3:');
   disp(P{3});
   disp('P4:');
   disp(P{4});
   disp('P5:');
   disp(P{5});
   disp('P6:');
   disp(P{6});
   disp('Q:');
   disp(Q);

   disp('and X = ⍰PₖQ(Pₖ)^T  Where k runs from 1 to 6, ')
   disp('Then what is the sum of diagonal elements of the matrix X?')
   fprintf('(a) %d (b) %d (c) %d (d) %d \n', jumbledOptions(1), jumbledOptions(2), jumbledOptions(3), jumbledOptions(4));



   % Display the correct option
   disp('Correct Option:')
   fprintf('Option (%c)\n', char(correctOptionIndex + 96));


end

% Explanation of the answer
   disp('Explanation:')
   disp('We need use the fact that Trace(AB) = Trace(BA)');
   disp('Trace(⍰PₖQ(Pₖ)^T ) = Trace(⍰Q(Pₖ)^TPₖ and We can see that (Pₖ)^TPₖ = (k^2)I');
   disp('Where I is identity matrix');
   disp('So Trace(⍰PₖQ(Pₖ)^T ) = (k^2)⍰Trace(Q)')
   disp('= 6*(k^2)*Trace(Q)');
```

```
Q2V1
 Let
P1:
     2     0     0
     0     2     0
     0     0     2
```

P2:
2 0 0
0 0 2
0 2 0
P3:
0 2 0
2 0 0
0 0 2
P4:
0 2 0
0 0 2
2 0 0
P5:
0 0 2
2 0 0
0 2 0
P6:
0 0 2
0 2 0
2 0 0
Q:
100.0000 11.0000 6.5000
11.0000 54.0000 12.0000
6.5000 12.0000 -86.0000
and $X = \Sigma P_k Q(P_k)^T$ Where k runs from 1 to 6,
Then what is the sum of diagonal elements of the matrix X?
(a) 7344 (b) 14688 (c) 29376 (d) 2448
Correct Option:
Option (b)

Q2V2
Let
P1:
2 0 0
0 2 0
0 0 2
P2:
2 0 0
0 0 2
0 2 0
P3:
0 2 0
2 0 0
0 0 2
P4:
0 2 0
0 0 2
2 0 0
P5:
0 0 2
2 0 0
0 2 0
P6:
0 0 2
0 2 0
2 0 0
Q:
66.0000 36.5000 -11.0000
36.5000 48.0000 27.0000
-11.0000 27.0000 12.0000
and $X = \Sigma P_k Q(P_k)^T$ Where k runs from 1 to 6,
Then what is the sum of diagonal elements of the matrix X?
(a) 54432 (b) 27216 (c) 13608 (d) 4536
Correct Option:
Option (b)

Q2V3
Let
P1:
1 0 0
0 1 0
0 0 1
P2:
1 0 0
0 0 1
0 1 0
P3:
0 1 0
1 0 0
0 0 1
P4:
0 1 0
0 0 1
1 0 0
P5:
0 0 1
1 0 0
0 1 0
P6:
0 0 1
0 1 0
1 0 0
Q:

```
   71.0000  -35.5000  -95.0000
  -35.5000   19.0000  -10.0000
  -95.0000  -10.0000  -54.0000
and X = ΣPₖQ(Pₖ)^T  Where k runs from 1 to 6,
Then what is the sum of diagonal elements of the matrix X?
(a) 15552 (b) 7776 (c) 1296 (d) 3888
Correct Option:
Option (b)

Q2V4
 Let
P1:
     6     0     0
     0     6     0
     0     0     6
P2:
     6     0     0
     0     0     6
     0     6     0
P3:
     0     6     0
     6     0     0
     0     0     6
P4:
     0     6     0
     0     0     6
     6     0     0
P5:
     0     0     6
     6     0     0
     0     6     0
P6:
     0     0     6
     0     6     0
     6     0     0
Q:
  -44.0000    9.5000   49.5000
    9.5000  -85.0000   33.0000
   49.5000   33.0000   99.0000
and X = ΣPₖQ(Pₖ)^T  Where k runs from 1 to 6,
Then what is the sum of diagonal elements of the matrix X?
(a) -1080 (b) -3240 (c) -6480 (d) -12960
Correct Option:
Option (c)

Q2V5
 Let
P1:
     4     0     0
     0     4     0
     0     0     4
P2:
     4     0     0
     0     0     4
     0     4     0
P3:
     0     4     0
     4     0     0
     0     0     4
P4:
     0     4     0
     0     0     4
     4     0     0
P5:
     0     0     4
     4     0     0
     0     4     0
P6:
     0     0     4
     0     4     0
     4     0     0
Q:
   42.0000   -3.0000  -21.5000
   -3.0000  -56.0000  -32.0000
  -21.5000  -32.0000  -86.0000
and X = ΣPₖQ(Pₖ)^T  Where k runs from 1 to 6,
Then what is the sum of diagonal elements of the matrix X?
(a) -3600 (b) -10800 (c) -21600 (d) -43200
Correct Option:
Option (c)
Explanation:
We need use the fact that Trace(AB) = Trace(BA)
Trace(ΣPₖQ(Pₖ)^T ) = Trace(ΣQ(Pₖ)^TPₖ and We can see that (Pₖ)^TPₖ = (k^2)I
Where I is identity matrix
So Trace(ΣPₖQ(Pₖ)^T ) = (k^2)ΣTrace(Q)
= 6*(k^2)*Trace(Q)
```

QUESTION 3 (OPTIMIZATION)

```
clear all
for i = 1:5
    %Title
```

```
fprintf('\nQ3V%d\n ',i);

syms x;
syms y;

a = randi([2, 5]);
b = randi([1, 6]);
c = randi([3, 7]);
d = randi([2, 8]);
e = randi([1, 7]);
f = randi([3, 9]);

%Create a function in x and y of form: ax^3 + bx^2 + cy^2 + dx + ey + f
% f(x, y) = (ax-b)^2 + cx^3 + (dy+e)^2 - f
coefficient_x3 = c;
coefficient_x2 = a*a;
coefficient_x = (-2)*a*b;
coefficient_y2 = d*d;
coefficient_y = 2*d*e;
constant_term = b*b + e*e - f;

%computing gradient of f
% Let grad(f) = [A ;B]
grad_f = [2*a*(a*x-b) + 3*c*x*x; 2*d*(d*y + e)];

%Finding stationary points, for which grad(f) = 0
%The points are (x1, y0) and (x2, y0)

x1 = (-a*a + sqrt(a^4 + 6*a*b*c)) / (3*c);
x2 = (-a*a - sqrt(a^4 + 6*a*b*c)) / (3*c);
y0 = -(e/d);

%Computing Hessian, H(x, y)
H = @(x) [2*a^2 + 6*c*x 0 ; 0 2*d^2];
H_disp = [2*a^2 + 6*c*x 0 ; 0 2*d^2];

% Define the 16 statements
statements_opt1 = {
    '(x1,y0) is a point of minima'
    '(x2,y0) is a saddle point'
    '(x1,y0) is a point of maxima'
    '(x1,y0) is a saddle point'
    '(x2,y0) is a point of minima'
    '(x2,y0) is a point of maxima'
};
% Randomly select 4 statements
selected_indices_opt1 = randperm(6, 4);
options_opt1 = char('a' + (1:numel(selected_indices_opt1)) - 1);
answer_key_opt1 = [1 1 0 0 0 0]; % Correctness of each statement
%Question
disp('Consider a 2-variable function f(x, y) defined as follows.');disp(' ');
fprintf('f(x,y) = %dx^3 + %dx^2 + %dy^2 - %dx + %dy + %d\n', coefficient_x3, ...
    coefficient_x2, coefficient_y2, -coefficient_x, coefficient_y, constant_term);
disp('Which of the following options are correct for the given function f(x, y)?')
fprintf('given points (x1, y0) = (%f , %f ) and (x2,y0) = (%f , %f) \n', x1, y0, x2, y0)

%Generate options
for i = 1:numel(selected_indices_opt1)
    disp([char(options_opt1(i)) ') ' statements_opt1{selected_indices_opt1(i)}])
end
disp(['e) None of the above'])

%correct options
% (x1, y0) is a point of minima
% (x2, y0) is a saddle point
% Check the correctness of the answer
    correct_options_opt1 = options_opt1(answer_key_opt1(selected_indices_opt1) == 1);


    % Display the answer

    if isempty(correct_options_opt1)
        disp('answer is e) None of the above')
    else
        disp(['The correct options are: ' char(correct_options_opt1)])

    end

%Explanation of the answer
disp(' '); disp('Explanation:');
disp('First we find the stationary points for f using ∇f = 0');
disp('The gradient of f is');
disp(grad_f);
fprintf('The roots of %dx^2 + %dx - %d = 0 are %f, %f\n', 3*c, 2*a*a, 2*a*b, x1, x2);
fprintf('The roots of %dy + %d = 0 is %d\n', 2*d*d, 2*d*e, y0);
fprintf('Therefore stationary points are (%f, %f) and (%f, %f)\n', x1,y0,x2,y0);
disp('The hessian of f is ');
disp(H_disp);disp('');
fprintf('The hessian of f at (%f, %f) is\n', x1, y0);
disp(H(x1));disp('');
fprintf('The eigen values are: %d, %d. So f is minimum at (%f, %f)\n', 2*a^2 + 6*c*x1,2*d^2,x1,y0);
fprintf('\nThe hessian of f at (%f, %f) is\n', x2, y0);
disp(H(x2));disp('');
```

```
    fprintf('The eigen values of f are: %d, %d. So, (%f, %f) is a saddle point\n',2*a^2 + 6*c*x2,2*d^2,x2,y0);

end
```

```
Q3V1
 Consider a 2-variable function f(x, y) defined as follows.

f(x,y) = 5x^3 + 9x^2 + 4y^2 - 18x + 4y + 6
Which of the following options are correct for the given function f(x, y)?
given points (x1, y0) = (0.649000 , -0.500000 ) and (x2,y0) = (-1.849000 , -0.500000)
a) (x1,y0) is a point of minima
b) (x1,y0) is a saddle point
c) (x2,y0) is a point of maxima
d) (x2,y0) is a saddle point
e) None of the above
The correct options are: ad

Explanation:
First we find the stationary points for f using ∇f = 0
The gradient of f is
15*x^2 + 18*x - 18
           8*y + 4
The roots of 15x^2 + 18x - 18 = 0 are 0.649000, -1.849000
The roots of 8y + 4 = 0 is -5.000000e-01
Therefore stationary points are (0.649000, -0.500000) and (-1.849000, -0.500000)
The hessian of f is
[30*x + 18, 0]
[        0, 8]
The hessian of f at (0.649000, -0.500000) is
   37.4700         0
         0    8.0000
The eigen values are: 3.746999e+01, 8. So f is minimum at (0.649000, -0.500000)

The hessian of f at (-1.849000, -0.500000) is
  -37.4700         0
         0    8.0000
The eigen values of f are: -3.746999e+01, 8. So, (-1.849000, -0.500000) is a saddle point

Q3V2
 Consider a 2-variable function f(x, y) defined as follows.

f(x,y) = 4x^3 + 16x^2 + 64y^2 - 24x + 96y + 37
Which of the following options are correct for the given function f(x, y)?
given points (x1, y0) = (0.610317 , -0.750000 ) and (x2,y0) = (-3.276984 , -0.750000)
a) (x1,y0) is a saddle point
b) (x2,y0) is a point of minima
c) (x1,y0) is a point of minima
d) (x1,y0) is a point of maxima
e) None of the above
The correct options are: c

Explanation:
First we find the stationary points for f using ∇f = 0
The gradient of f is
12*x^2 + 32*x - 24
        128*y + 96
The roots of 12x^2 + 32x - 24 = 0 are 0.610317, -3.276984
The roots of 128y + 96 = 0 is -7.500000e-01
Therefore stationary points are (0.610317, -0.750000) and (-3.276984, -0.750000)
The hessian of f is
[24*x + 32,   0]
[        0, 128]
The hessian of f at (0.610317, -0.750000) is
   46.6476         0
         0  128.0000
The eigen values are: 4.664762e+01, 128. So f is minimum at (0.610317, -0.750000)

The hessian of f at (-3.276984, -0.750000) is
  -46.6476         0
         0  128.0000
The eigen values of f are: -4.664762e+01, 128. So, (-3.276984, -0.750000) is a saddle point

Q3V3
 Consider a 2-variable function f(x, y) defined as follows.

f(x,y) = 7x^3 + 9x^2 + 4y^2 - 36x + 12y + 36
Which of the following options are correct for the given function f(x, y)?
given points (x1, y0) = (0.949093 , -1.500000 ) and (x2,y0) = (-1.806236 , -1.500000)
a) (x1,y0) is a point of minima
b) (x1,y0) is a point of maxima
c) (x2,y0) is a point of maxima
d) (x2,y0) is a saddle point
e) None of the above
The correct options are: ad

Explanation:
First we find the stationary points for f using ∇f = 0
The gradient of f is
21*x^2 + 18*x - 36
          8*y + 12
The roots of 21x^2 + 18x - 36 = 0 are 0.949093, -1.806236
```

```
The roots of 8y + 12 = 0 is -1.500000e+00
Therefore stationary points are (0.949093, -1.500000) and (-1.806236, -1.500000)
The hessian of f is
[42*x + 18, 0]
[        0, 8]
The hessian of f at (0.949093, -1.500000) is
   57.8619         0
         0    8.0000
The eigen values are: 5.786190e+01, 8. So f is minimum at (0.949093, -1.500000)

The hessian of f at (-1.806236, -1.500000) is
  -57.8619         0
         0    8.0000
The eigen values of f are: -5.786190e+01, 8. So, (-1.806236, -1.500000) is a saddle point

Q3V4
 Consider a 2-variable function f(x, y) defined as follows.

f(x,y) = 4x^3 + 4x^2 + 16y^2 - 20x + 32y + 37
Which of the following options are correct for the given function f(x, y)?
given points (x1, y0) = (1.000000 , -1.000000 ) and (x2,y0) = (-1.666667 , -1.000000)
a) (x2,y0) is a saddle point
b) (x1,y0) is a point of minima
c) (x2,y0) is a point of minima
d) (x2,y0) is a point of maxima
e) None of the above
The correct options are: ab

Explanation:
First we find the stationary points for f using ∇f = 0
The gradient of f is
12*x^2 + 8*x - 20
        32*y + 32
The roots of 12x^2 + 8x - 20 = 0 are 1.000000, -1.666667
The roots of 32y + 32 = 0 is -1
Therefore stationary points are (1.000000, -1.000000) and (-1.666667, -1.000000)
The hessian of f is
[24*x + 8,  0]
[       0, 32]
The hessian of f at (1.000000, -1.000000) is
    32     0
     0    32
The eigen values are: 32, 32. So f is minimum at (1.000000, -1.000000)

The hessian of f at (-1.666667, -1.000000) is
   -32     0
     0    32
The eigen values of f are: -32, 32. So, (-1.666667, -1.000000) is a saddle point

Q3V5
 Consider a 2-variable function f(x, y) defined as follows.

f(x,y) = 5x^3 + 16x^2 + 4y^2 - 48x + 28y + 78
Which of the following options are correct for the given function f(x, y)?
given points (x1, y0) = (1.016067 , -3.500000 ) and (x2,y0) = (-3.149400 , -3.500000)
a) (x1,y0) is a point of maxima
b) (x2,y0) is a saddle point
c) (x1,y0) is a point of minima
d) (x2,y0) is a point of maxima
e) None of the above
The correct options are: bc

Explanation:
First we find the stationary points for f using ∇f = 0
The gradient of f is
15*x^2 + 32*x - 48
          8*y + 28
The roots of 15x^2 + 32x - 48 = 0 are 1.016067, -3.149400
The roots of 8y + 28 = 0 is -3.500000e+00
Therefore stationary points are (1.016067, -3.500000) and (-3.149400, -3.500000)
The hessian of f is
[30*x + 32, 0]
[        0, 8]
The hessian of f at (1.016067, -3.500000) is
   62.4820         0
         0    8.0000
The eigen values are: 6.248200e+01, 8. So f is minimum at (1.016067, -3.500000)

The hessian of f at (-3.149400, -3.500000) is
  -62.4820         0
         0    8.0000
The eigen values of f are: -6.248200e+01, 8. So, (-3.149400, -3.500000) is a saddle point
```

QUESTION 4 (OPTIMIZATION)

```
clear all

for i=1:5
    %Title
    fprintf('\nQ4V%d\n', i);

    % Create the function
```

```
f = @(x, y) a*x*x + b*y*y;

%Set inital guess
x_int = 1;
y_int = 1;

%Generating random values for question; more than 500 variants
%such that answer lies in [1, 4]
a = randi([1,5]);
b = a;

if (a==1)
    alpha = 0.41 + 0.01*randi([-7, 8]); % Learning rate
    tolerance = 0.075+0.001*randi([-5,5]); % tolerance on ∇f
end

if (a==2)
    alpha = 0.21+0.01*randi([-3, 4]); % Learning rate
    tolerance = 0.075+0.001*randi([-5,5]); % tolerance on ∇f
end

if (a==3)
    alpha = 0.14+0.01*randi([-2, 3]); % Learning rate
    tolerance = 0.055+0.001*randi([-5,5]); % tolerance on ∇f
end

if (a==4)
    alpha = 0.10+0.01*randi([-1, 2]); % Learning rate
    tolerance = 0.055+0.001*randi([-5,5]); % tolerance on ∇f
end

if (a==5)
    alpha = 0.09+0.001*randi([-10, 8]); % Learning rate
    tolerance = 0.025+0.001*randi([-5,5]); % tolerance on ∇f
end

%computing gradient of f
grad_f_x = @(x) 2*a*x; %x-component of gradient
grad_f_y = @(y) 2*b*y; %y-component of gradient

gradf_x = grad_f_x(1); %At the start
gradf_y = grad_f_y(1);

count = 0; %No. of iterations
%Start performing the iterations

x_temp = x_int;
y_temp = y_int;
while ((gradf_x > tolerance) && (gradf_y > tolerance))

    x = x_temp - alpha*gradf_x;
    y = y_temp - alpha*gradf_y; %compute points

    gradf_x = grad_f_x(x);
    gradf_y = grad_f_y(y);

    x_temp = x; %Updating the kth point of iteration
    y_temp = y;

    count = count + 1; % incrementing the count
end

% Define the options
option1_opt2 = count; %correct answer
option2_opt2 = count+1;
option3_opt2 = count+3;
option4_opt2 = count-1;

options_opt2 = [option1_opt2, option2_opt2, option3_opt2, option4_opt2];


% Generate random permutation of array indices
% We use this for jumbling the options
permutedIndices_opt2 = randperm(numel(options_opt2));

% Ensuring the options are jumbled each time
jumbledOptions_opt2 = options_opt2(permutedIndices_opt2);

% Determine the correct option index

correctOptionIndex_opt2 = find(jumbledOptions_opt2 == count);

%Question
disp('');disp('Suppose an ant is gracing on a surface and the surface is approximated by the function');
fprintf('f(x,y) = %dx^2 + %dy^2\n',a,b);
fprintf('The ant is moving on the surface to reach an optimum point. Initially, starting from the point (%d,%d).\n',x_int,y_int);
disp('The movement of ant is in accordance with the steepest decent algorithm learned.');
disp('Find out the number of times the ant changes the place from one point to another.');
fprintf('The ant changes the place(point) until the tolerance of %.3f is obtained on the steepest gradient taken by it at that point\n', tolerance);
fprintf('Use the leraning factor as %.2f\n\n', alpha);

%Options
disp('Options:');
```

```
    fprintf('(a) %d\n', jumbledOptions_opt2(1));
    fprintf('(b) %d\n', jumbledOptions_opt2(2));
    fprintf('(c) %d\n', jumbledOptions_opt2(3));
    fprintf('(d) %d\n\n', jumbledOptions_opt2(4));


    %Correct answer - count

    % Display the correct option
    fprintf('Correct Option : Option (%c)\n\n', char(correctOptionIndex_opt2 + 96));

    %Explanation
    disp('Explanation:');disp('');
    disp('The steepest decent algorithm states that x_{k+1} = x_{k} - α.∇f|_{xk,yk}');disp('');
    disp('Using that algorithm we get the iterated points as:');disp('');


    %Print the nth iterated values
    gradf_x = grad_f_x(1); %At the start
    gradf_y = grad_f_y(1);
    temp = 0;
    x0 = 1;
    y0 = 1;
    while (temp ~= count)

        x = x0 - alpha*gradf_x;
        y = y0 - alpha*gradf_y; %compute points

        gradf_x = grad_f_x(x);
        gradf_y = grad_f_y(y);

        x0 = x; %Updating the kth point of iteration
        y0 = y;
        temp = temp + 1; % incrementing the count
        fprintf('%dth iteration gives (%f,%f) and ∇f at that point is (%f,%f)\n',temp,x,y,gradf_x,gradf_y);
    end
    fprintf('We see that at this point we reached the tolerance for ∇f i.e, %.3f\n', tolerance);
    fprintf('Hence, number of iterations is %d\n', count);
end
```

```
Q4V1
Suppose an ant is gracing on a surface and the surface is approximated by the function
f(x,y) = 2x^2 + 2y^2
The ant is moving on the surface to reach an optimum point. Initially, starting from the point (1,1).
The movement of ant is in accordance with the steepest decent algorithm learned.
Find out the number of times the ant changes the place from one point to another.
The ant changes the place(point) until the tolerance of 0.076 is obtained on the steepest gradient taken by it at that point
Use the leraning factor as 0.18

Options:
(a) 4
(b) 3
(c) 7
(d) 5

Correct Option : Option (a)

Explanation:
The steepest decent algorithm states that x_{k+1} = x_{k} - α.∇f|_{xk,yk}
Using that algorithm we get the iterated points as:
1th iteration gives (0.280000,0.280000) and ∇f at that point is (1.120000,1.120000)
2th iteration gives (0.078400,0.078400) and ∇f at that point is (0.313600,0.313600)
3th iteration gives (0.021952,0.021952) and ∇f at that point is (0.087808,0.087808)
4th iteration gives (0.006147,0.006147) and ∇f at that point is (0.024586,0.024586)
We see that at this point we reached the tolerance for ∇f i.e, 0.076
Hence, number of iterations is 4

Q4V2
Suppose an ant is gracing on a surface and the surface is approximated by the function
f(x,y) = 5x^2 + 5y^2
The ant is moving on the surface to reach an optimum point. Initially, starting from the point (1,1).
The movement of ant is in accordance with the steepest decent algorithm learned.
Find out the number of times the ant changes the place from one point to another.
The ant changes the place(point) until the tolerance of 0.022 is obtained on the steepest gradient taken by it at that point
Use the leraning factor as 0.08

Options:
(a) 4
(b) 5
(c) 3
(d) 7

Correct Option : Option (a)

Explanation:
The steepest decent algorithm states that x_{k+1} = x_{k} - α.∇f|_{xk,yk}
Using that algorithm we get the iterated points as:
1th iteration gives (0.170000,0.170000) and ∇f at that point is (1.700000,1.700000)
2th iteration gives (0.028900,0.028900) and ∇f at that point is (0.289000,0.289000)
3th iteration gives (0.004913,0.004913) and ∇f at that point is (0.049130,0.049130)
4th iteration gives (0.000835,0.000835) and ∇f at that point is (0.008352,0.008352)
```

```
We see that at this point we reached the tolerance for ∇f i.e, 0.022
Hence, number of iterations is 4

Q4V3
Suppose an ant is gracing on a surface and the surface is approximated by the function
f(x,y) = 4x^2 + 4y^2
The ant is moving on the surface to reach an optimum point. Initially, starting from the point (1,1).
The movement of ant is in accordance with the steepest decent algorithm learned.
Find out the number of times the ant changes the place from one point to another.
The ant changes the place(point) until the tolerance of 0.051 is obtained on the steepest gradient taken by it at that point
Use the leraning factor as 0.11

Options:
(a) 4
(b) 2
(c) 3
(d) 6

Correct Option : Option (c)

Explanation:
The steepest decent algorithm states that x_{k+1} = x_{k} - α.∇f|_{xk,yk}
Using that algorithm we get the iterated points as:
1th iteration gives (0.120000,0.120000) and ∇f at that point is (0.960000,0.960000)
2th iteration gives (0.014400,0.014400) and ∇f at that point is (0.115200,0.115200)
3th iteration gives (0.001728,0.001728) and ∇f at that point is (0.013824,0.013824)
We see that at this point we reached the tolerance for ∇f i.e, 0.051
Hence, number of iterations is 3

Q4V4
Suppose an ant is gracing on a surface and the surface is approximated by the function
f(x,y) = 5x^2 + 5y^2
The ant is moving on the surface to reach an optimum point. Initially, starting from the point (1,1).
The movement of ant is in accordance with the steepest decent algorithm learned.
Find out the number of times the ant changes the place from one point to another.
The ant changes the place(point) until the tolerance of 0.029 is obtained on the steepest gradient taken by it at that point
Use the leraning factor as 0.08

Options:
(a) 7
(b) 4
(c) 3
(d) 5

Correct Option : Option (b)

Explanation:
The steepest decent algorithm states that x_{k+1} = x_{k} - α.∇f|_{xk,yk}
Using that algorithm we get the iterated points as:
1th iteration gives (0.200000,0.200000) and ∇f at that point is (2.000000,2.000000)
2th iteration gives (0.040000,0.040000) and ∇f at that point is (0.400000,0.400000)
3th iteration gives (0.008000,0.008000) and ∇f at that point is (0.080000,0.080000)
4th iteration gives (0.001600,0.001600) and ∇f at that point is (0.016000,0.016000)
We see that at this point we reached the tolerance for ∇f i.e, 0.029
Hence, number of iterations is 4

Q4V5
Suppose an ant is gracing on a surface and the surface is approximated by the function
f(x,y) = 1x^2 + 1y^2
The ant is moving on the surface to reach an optimum point. Initially, starting from the point (1,1).
The movement of ant is in accordance with the steepest decent algorithm learned.
Find out the number of times the ant changes the place from one point to another.
The ant changes the place(point) until the tolerance of 0.075 is obtained on the steepest gradient taken by it at that point
Use the leraning factor as 0.49

Options:
(a) 2
(b) 4
(c) 0
(d) 1

Correct Option : Option (d)

Explanation:
The steepest decent algorithm states that x_{k+1} = x_{k} - α.∇f|_{xk,yk}
Using that algorithm we get the iterated points as:
1th iteration gives (0.020000,0.020000) and ∇f at that point is (0.040000,0.040000)
We see that at this point we reached the tolerance for ∇f i.e, 0.075
Hence, number of iterations is 1
```

QUESTION 5 (STATISTICS)

```
clearvars;  format compact;

for i = 1:5
    %Title
    fprintf('\nQ5V%d\n ',i);
    %Values are choosen so as to ensure the question is hand-solvable
    % Total number of balls
    n = randi([5,15]);
    %Getting random integer coefficients for the linear inequality
    k = randi([1,n]);%constant term in the inequality
```

```
    p = randi([1,3]);%coefficient of x
    q = randi([p,4]);%coefficient of y

    % Initialize the counter for satisfying cases
    count = 0;

    % Iterate over all possible values of x and y
    for x = 1:n
        for y = 1:n
            % Check if the inequality holds
            if (p*x- q*y + k) > 0
                % Increment the counter
                count = count + 1;
            end
        end
    end

    % Calculate the probability
    probability = count / (n * n);
    %rounding off to 3 decimal places
    probability = round(probability, 3);

    % Generate three random values in the range [0, 1]
    % We use these as options
    options_prob1 = rand(1, 3);

    % Round off the options to three decimal places
    rounded_options = round(options_prob1, 3);
    rounded_options = [rounded_options probability];

    % Generate random permutation of array indices
    % We use this for jumbling the options
    permutedIndices_prob1 = randperm(numel(rounded_options));

    % Ensuring the options are jumbled each time
    jumbledOptions_prob1 = rounded_options(permutedIndices_prob1);


    % Determine the correct option index
    correctAnswer = probability;
    correctOptionIndex_prob1 = find(jumbledOptions_prob1 == correctAnswer);

    %The Question statement
    fprintf('Assume there are %d balls in a bag of same size and color,', n);
    fprintf(' numbered 1,2,....,%d, were put into a packet. ', n);
    disp('Now Arjun draws a ball from the packet, noted that it is of number "x" and puts back it.');
    disp('Then Bunny also draws a ball from the packet and noted that it is of number "y".');
    fprintf('Then the probability for the inequality %d*x - %d*y + %d > 0 to hold', p,q,k);
    disp('(Round off answer to 3 decimal places)');
    fprintf('\n');

    %Displaying the options
    fprintf('(a) %.3f (b) %.3f (c) %.3f (d) %.3f \n\n\n', jumbledOptions_prob1(1), jumbledOptions_prob1(2), jumbledOptions_prob1(3), jumbledOptions_prob1(4));

    % Display the correct option
    fprintf('Correct Option : Option (%c)\n\n', char(correctOptionIndex_prob1 + 96));


end

% Explanation of the answer
    disp('Explanation:')
    disp('Let us take an example problem and solve...All can be solved in the similar way');
    disp(' Let us take the case of x-2y+10 > 0 and there are 9 balls ');
    disp('The total number of possible events is 9*9 = 81');
    disp('From x-2y+10 > 0 we get 2y < x+10.');
    disp('We find that when y=1,2,3,4,5, x can take any value in 1,2,..,9 to make the ineuality hold. Then we have 9*5 = 45 admissible events.' );
    disp('When y = 6, x can be 3,4,..,9 and there are 7 admissible events.');
    disp('When y = 7, x can be 5,6..,9 and there are 5 admissible events.');
    disp('When y = 8, x can be 7,8,9 and there are 3 admissble events.');
    disp('When y = 9, x can be 9 and there is 1 admissible event.');
    disp('So the required probability is (45+7+5+3+1)/81 = 61/81');
    disp('Any question of similar model can be done in the similar way');
```

```
Q5V1
 Assume there are 10 balls in a bag of same size and color, numbered 1,2,....,10, were put into a packet. Now Arjun draws a ball from the packet, noted that it
Then Bunny also draws a ball from the packet and noted that it is of number "y".
Then the probability for the inequality 1*x - 1*y + 5 > 0 to hold(Round off answer to 3 decimal places)

(a) 0.850 (b) 0.606 (c) 0.645 (d) 0.182


Correct Option : Option (a)


Q5V2
 Assume there are 15 balls in a bag of same size and color, numbered 1,2,....,15, were put into a packet. Now Arjun draws a ball from the packet, noted that it
Then Bunny also draws a ball from the packet and noted that it is of number "y".
Then the probability for the inequality 3*x - 3*y + 10 > 0 to hold(Round off answer to 3 decimal places)

(a) 0.707 (b) 0.591 (c) 0.510 (d) 0.900
```

```
Correct Option : Option (a)


Q5V3
 Assume there are 11 balls in a bag of same size and color, numbered 1,2,....,11, were put into a packet. Now Arjun draws a ball from the packet, noted that it
Then Bunny also draws a ball from the packet and noted that it is of number "y".
Then the probability for the inequality 2*x - 2*y + 9 > 0 to hold(Round off answer to 3 decimal places)

(a) 0.589 (b) 0.268 (c) 0.826 (d) 0.925


Correct Option : Option (c)


Q5V4
 Assume there are 11 balls in a bag of same size and color, numbered 1,2,....,11, were put into a packet. Now Arjun draws a ball from the packet, noted that it
Then Bunny also draws a ball from the packet and noted that it is of number "y".
Then the probability for the inequality 3*x - 3*y + 9 > 0 to hold(Round off answer to 3 decimal places)

(a) 0.973 (b) 0.702 (c) 0.076 (d) 0.425


Correct Option : Option (b)


Q5V5
 Assume there are 12 balls in a bag of same size and color, numbered 1,2,....,12, were put into a packet. Now Arjun draws a ball from the packet, noted that it
Then Bunny also draws a ball from the packet and noted that it is of number "y".
Then the probability for the inequality 3*x - 3*y + 3 > 0 to hold(Round off answer to 3 decimal places)

(a) 0.542 (b) 0.726 (c) 0.960 (d) 0.634


Correct Option : Option (a)

Explanation:
Let us take an example problem and solve...All can be solved in the similar way
 Let us take the case of x-2y+10 > 0 and there are 9 balls
The total number of possible events is 9*9 = 81
From x-2y+10 > 0 we get 2y < x+10.
We find that when y=1,2,3,4,5, x can take any value in 1,2,..,9 to make the ineuality hold. Then we have 9*5 = 45 admissible events.
When y = 6, x can be 3,4,..,9 and there are 7 admissible events.
When y = 7, x can be 5,6..,9 and there are 5 admissible events.
When y = 8, x can be 7,8,9 and there are 3 admissble events.
When y = 9, x can be 9 and there is 1 admissible event.
So the required probability is (45+7+5+3+1)/81 = 61/81
Any question of similar model can be done in the similar way
```

QUESTION 6 (STATISTICS)

```
clearvars;  format compact;
for i = 1:5
    %Title
    fprintf('\nQ6V%d\n ',i);
    % Define the winnings and losses
    m = randi([1, 10]); % Winnings for two tails
    n = randi([1, 10]); % Winnings for two heads
    p = randi([1, 10]); % Loss for one head and one tail

    % Generate random values for a and b between 0 and 1
    a = rand();
    b = rand();

    % Calculate the expected winnings
    E = (1 - a) * (1 - b) * m + a * b * n + (1 - a) * b * (-p) + a * (1 - b) * (-p);

    % Calculate the variance
    V = (1 - a) * (1 - b) * (m - E)^2 + a * b * (n - E)^2 + (1 - a) * b * (-p - E)^2 + a * (1 - b) * (-p - E)^2;

    % Calculate the ratio V/E
    ratio = V / E;

    % Define the options
    option1 = ratio;
    option2 = ratio*2 + ratio/2;
    option3 = ratio*2- ratio/2;
    option4 = ratio*0.65;

    options_prob2 = [option1, option2, option3, option4];
    options_prob2 = round(options_prob2, 4);

    % Generate random permutation of array indices
    % We use this for jumbling the options
    permutedIndices_prob2 = randperm(numel(options_prob2));

    % Ensuring the options are jumbled each time
    jumbledOptions_prob2 = options_prob2(permutedIndices_prob2);

    % Determine the correct option index
    ratio = round(ratio, 4);
```

```
    correctOptionIndex_prob2 = find(jumbledOptions_prob2 == ratio);


    % Display the question and options
    fprintf('You and a friend play a game where you each toss a coin. The coin is biased. The probability of getting a head on the first coin is %f and on the
    fprintf('If the upper faces on the coins are both tails, you win $%d; if the faces are both heads, you win $%d; if one shows a head and the other a tail, y
    disp('Your expected winnings is E and variance is V');
    disp('Then what is the V/E (rounded off to 4 decimal places)');
    disp('Options:');
    fprintf('(a) %.4f\n', jumbledOptions_prob2(1));
    fprintf('(b) %.4f\n', jumbledOptions_prob2(2));
    fprintf('(c) %.4f\n', jumbledOptions_prob2(3));
    fprintf('(d) %.4f\n', jumbledOptions_prob2(4));



    % Display the correct option
    fprintf('Correct Option : Option (%c)\n\n', char(correctOptionIndex_prob2 + 96));


end

 % Explanation of the answer
    disp('Explanation:')
    disp('If the probabilities of getting a head for one coin are denoted as a and for the other coin as b, we can adjust the calculations for the expected wir
    disp('E = ((1 - a) * (1 - b) * m) + (a * b * n) + ((1 - a) * b * (-p)) + (a * (1 - b) * (-p));')
    disp('V = ((1 - a) * (1 - b) * (m - E)^2) + (a * b * (n - E)^2) + ((1 - a) * b * (-p - E)^2) + (a * (1 - b) * (-p - E)^2);')
    disp('m is winnings for two tails, n is winnings for two heads p is loss for one head and one tail');
```

```
Q6V1
 You and a friend play a game where you each toss a coin. The coin is biased. The probability of getting a head on the first coin is 0.665270 and on the seconc
If the upper faces on the coins are both tails, you win $7; if the faces are both heads, you win $4; if one shows a head and the other a tail, you lose $5.
Your expected winnings is E and variance is V
Then what is the V/E (rounded off to 4 decimal places)
Options:
(a) 21.5223
(b) 33.1113
(c) 49.6669
(d) 82.7782
Correct Option : Option (b)


Q6V2
 You and a friend play a game where you each toss a coin. The coin is biased. The probability of getting a head on the first coin is 0.678951 and on the seconc
If the upper faces on the coins are both tails, you win $4; if the faces are both heads, you win $2; if one shows a head and the other a tail, you lose $9.
Your expected winnings is E and variance is V
Then what is the V/E (rounded off to 4 decimal places)
Options:
(a) -21.2235
(b) -9.1969
(c) -35.3725
(d) -14.1490
Correct Option : Option (d)


Q6V3
 You and a friend play a game where you each toss a coin. The coin is biased. The probability of getting a head on the first coin is 0.490251 and on the seconc
If the upper faces on the coins are both tails, you win $10; if the faces are both heads, you win $7; if one shows a head and the other a tail, you lose $9.
Your expected winnings is E and variance is V
Then what is the V/E (rounded off to 4 decimal places)
Options:
(a) 207.6169
(b) 138.4113
(c) 89.9673
(d) 346.0282
Correct Option : Option (b)


Q6V4
 You and a friend play a game where you each toss a coin. The coin is biased. The probability of getting a head on the first coin is 0.428237 and on the seconc
If the upper faces on the coins are both tails, you win $2; if the faces are both heads, you win $4; if one shows a head and the other a tail, you lose $2.
Your expected winnings is E and variance is V
Then what is the V/E (rounded off to 4 decimal places)
Options:
(a) 9.8873
(b) 22.8169
(c) 15.2113
(d) 38.0281
Correct Option : Option (c)


Q6V5
 You and a friend play a game where you each toss a coin. The coin is biased. The probability of getting a head on the first coin is 0.168217 and on the seconc
If the upper faces on the coins are both tails, you win $9; if the faces are both heads, you win $6; if one shows a head and the other a tail, you lose $3.
Your expected winnings is E and variance is V
Then what is the V/E (rounded off to 4 decimal places)
Options:
(a) 5.1893
(b) 3.4595
(c) 2.2487
(d) 8.6488
```

```
Correct Option : Option (b)

Explanation:
If the probabilities of getting a head for one coin are denoted as a and for the other coin as b, we can adjust the calculations for the expected winnings and
E = ((1 - a) * (1 - b) * m) + (a * b * n) + ((1 - a) * b * (-p)) + (a * (1 - b) * (-p));
V = ((1 - a) * (1 - b) * (m - E)^2) + (a * b * (n - E)^2) + ((1 - a) * b * (-p - E)^2) + (a * (1 - b) * (-p - E)^2);
m is winnings for two tails, n is winnings for two heads p is loss for one head and one tail
```