```
clc;
clear all;
fprintf("CH5019 - Project \nGroup no. - 25");
% below loops just prints all variants of all questions
% all questions are contained in functions that are define below
for i = 1:5
    q1(1,i);
end
for i = 1:5
    q2(2,i);
end
for i = 1:5
    q3(3,i);
end
for i = 1:5
    q4(4,i);
end
for i = 1:5
    q5(5,i);
end
for i = 1:5
    q6(6,i);
end

% to print the variant and question number
function variant(x,y)
    fprintf("\n----------------------------------------");
 fprintf("-------------------------------------------\n\n");
    if y == 1
        fprintf("Q %d. - Type: Numerical\n",x);
    end
    fprintf("<Q. %d, V. %d>\n\n",x,y);
end



% ===============================================   QUESTION - 1
 ===============================================
function q1(ques,vari)
    variant(ques,vari);
    %random input population data
    input_matrix=[31 33 17 19;27 32 22 19;24 30 32 14;26 27 29 18];
    %selection of one row
    pop_percent=randomgenerator(input_matrix);
    %random normaliser for rate constant
    rate_normaliser=randi([8,15]);
    % poisson dist.
    x=0;
    P=0;
    k=0;
    lambda=0;
    syms x  P k lambda;
```

```
    % probability of k events occuring at a given time interval x
    P(k,x,lambda)=exp(-lambda*x)*(lambda*x)^k/factorial(k);
    % k no. of events
    % x no.test cases(time interval in this case)
    %lambda=poisson ratio

    poisson_ratio=pop_percent.*rate_normaliser/100;
    a=poisson_ratio(1);
    b=poisson_ratio(2);
    c=poisson_ratio(3);
    d=poisson_ratio(4);

 answer=double(P(7,1,a)*P(3,1,b)*P(2,1,c)*P(1,1,d))+double(P(6,1,a)*P(4,1,b)*P(2,1

    %question
    fprintf("In a city X, there is a 24/7 vaccination center where the
 arrival of people follows a poison distribution.");
    fprintf('Assume a typical demographic distribution of \n')
    fprintf('\n     Number     Age Group\n\t%d :-\t0-18 \n\t%d :-
\t18-45 \n\t%d :-\t45-60 \n\t%d :-\tabove 60\n\n',pop_percent)
    fprintf('The arrival rate of each age group at the centre is
 proportional to their population percentage, and the sum of arrival
 rates is proportional to %d.\n',rate_normaliser)
    fprintf("Given that vaccines are delivered at the rate of 13/
hour , find the probability that in the duration of 1 hour, the more
 people from the older population get vaccinated than the younger
 ones.\n\n");
    fprintf("Options:\n");

    %answergenerator
    random_answer_matrix=[answer,answer+0.155,answer+0.355,answer
+0.1];
    tags=['A','B','C','D'];
    ordered_matrix=random_answer_matrix(randperm(4));

    %answer output
    for i=1:length(tags)
        fprintf('%s. %.3f\n',tags(i),ordered_matrix(i))
        if ordered_matrix(i)==answer
            number=i;
        end
    end

    fprintf('\nAnswer:%s \n',tags(number))
    fprintf('Explanation:\n\n');
    %explanation
    fprintf('Probablity of k possibilities in a time limit of x with a
 poisson ratio of lambda is')
    probability=P(k,x,lambda)
    fprintf('For a total of 13 vaccines per hour the possible
 permutations are:-    ')
    fprintf('(in the descending order of age groups)')
    permutations=[7,3,2,1;6,4,2,1;5,4,3,1]
```

```
    fprintf('The total probability is the sum of all these
 possibilities\n= %.3f\n',answer)

        %generator function
        function pop_percent=randomgenerator(input_matrix)
            pop_percent=input_matrix(randi(4),:);
        end
end



% ===============================================   QUESTION - 2
 ===============================================
function q2(ques,vari)
    variant(ques,vari);
    n=15;
    fprintf("You are the regional manager of a famous paper selling
 company, and make sales over a period of %d days. The profits
 for each day automatically gets parsed to a specific software.
 The corporate expects you to have made profits in an increasing
 order.Unfortunately, the sort function in that software is faulty and
 doesn't always yield the right answer.\n",n);
    fprintf("The sort function in this software performs sequential
 operations of the type [D_i ,P_i], which means that the profits-list
 from indices (days) [1,D_i] would be sorted with a probability of
 P_i , or would remain the same with a probability of 1-P_i .\n");
    profit=get_profit(n);
    fprintf("\nGiven, profits during the given period:   [");
    for i = 1:n
        cur=round(profit(i),1);
        fprintf("%d ",cur);
    end
    fprintf("]\n\nFind the probability that the profits' list would be
 sorted after performing ALL of the below operations.\n")
    fprintf("Sequential operations: (of the form [D_i , P_i])\n");
    store=get_store(n);
    for i = 1:n
        cur=round(store(i,1));
        fprintf("[ %2d , %.4f ]\n",cur,store(i,2));
    end
    fprintf("\nOptions\n");
    Id=["A.","B.","C.","D."];
    options=solve(profit,store);
    for i = 1:4
        cur=round(options(i),4);
        fprintf("%s  %.4f \n",Id(i),cur);
    end
    fprintf("\nAnswer: D\n");
    fprintf("Explanation:\n");
    fprintf("Firstly, we make the actual sorted profits array and
 compare it with the given array.\n\n\tConsider 'idx' as the largest
 index such that profits[idx] != sorted_profits[idx] holds. (which
 in this case is %d).\n\nSo, we are not interested in the operations
 with D_i less than idx, since the array will still be unsorted. Now,
```

```
let us look at the case where we *never* get a sorted array. The
probability for that to happen is product of all (1-P_i)'s for every
i>=idx .\n\nThe final answer is 1 - (the above result) , that is, 1 -
(product of all (1-P_i)'s') for every i >= idx\n",options(5));
   fprintf("So, the probability will be :\n\n= 1 - ");
   for i = options(5):n
       fprintf("(1 - %f)",store(i,2));
       if i ~= n
           fprintf("*");
       end
   end
   fprintf("\n= %.4f \n",options(4));
   % Function to generate profits array, such that idx is some value
not equal to n
   function a=get_profit(n)
       a=zeros(1,n);
       for i = 1:n
           a(i)=randi(999);
       end
       idx=3+randi(2);
       a=sort(a);
       tmp=zeros(0,0);
       for i = n-idx+1:n
           tmp(i-n+idx)=a(i);
       end
       for i = n-idx+1:n
           a(end)=[];
       end
       shuffle_index=randperm(length(a));
       b=a;
       for i = 1:length(a)
           a(i)=b(shuffle_index(i));
       end
       a=[a tmp];
   end
   % Function to generate operations array for days numbered 1 to n
   function a=get_store(n)
       a=zeros(n,2);
       for i = 1:n
           a(i,1)=i;
       end
       for i = 1:n
           r=randi(999);
           r=min(r,1000-r);
           a(i,2)=round(r/1000,4);
       end
   end
   % Solver function
   function res=solve(profit,store)
       n=length(profit);
       b=sort(profit);
       for i = n:-1:1
           if profit(i) ~= b(i)
               idx=i;
```

```
                break;
            end
        end
        p=1;
        for i = idx:n
            p=p*(1-store(i,2));
        end
        p=1-p;
        p=round(p,5);
        random_numbers=randperm(round(p,3)*1000-1);
        random_numbers=random_numbers(1:4);
        res=zeros(1,4);
        for i = 1:3
            res(i)=random_numbers(i)/1000;
        end
        res(4)=p;
        res(5)=idx;
    end
end



% ===============================================   QUESTION - 3
 ===============================================
function q3(ques,vari)
    variant(ques,vari);
    initial=zeros(1,3);
    for i = 1:3
        initial(i)=-10+randi(20);
    end
    angle_degs=15*(randi(11));
    angle=angle_degs*pi/180;
    final=zeros(1,3);
    for i = 1:3
        final(i)=-10+randi(20);
    end
    fprintf("A housefly in a room, travels to the point (%d,
 %d, %d), and disorients its path by an angle of %d degrees
 with respect to the positive X-axis, and finally reaches
 the point P(%d, %d, %d) with respect to this new frame.
\n",initial(1),initial(2),initial(3),angle_degs,final(1),final(2),final(3));
    fprintf("Let this final coordinate, when read by the initial
 coordinate frame, be Q(x,y,z). ");
    fprintf("Which of the following matrices has eigenvalues equal to
 the coordinates of Q?\n")
    fprintf("(Consider the starting point of the housefly as the
 origin in the initial frame)\n\n");
    fprintf("\n\tOptions:\n");
    Id=["A.","B.","C.","D."];

    rot=get_rot(angle);
    trans=get_trans(initial);
    AB=get_AB(rot,trans);
    result=get_result(AB,final);
```

```
for i = 1:4
    now=zeros(1,3);
    for j = 1:3
        now(j)=-5+randi(10)+randi(9999)/10000;
    end
    if i == 4
        now=result;
    end
    now_diag=diag(now);
    r=rand(3); %random_matrix
    % option matrix = inv(r)*diag(eig_values)*r
    final=inv(r)*now_diag*r;
    fprintf("%s\n",Id(i));
    disp(final);
end
fprintf("\nAnswer: D.\nExplanation:\n\n");
fprintf("Let us name the initial frame as A, and the final frame
(after translation and rotation) as B.\n");
fprintf("Firstly we formulate the rotation matrix.\nIt is given
as: (for an angle 'x' , wrt Positive X-axis)\n\n");
fprintf("\t1     0      0      0  \n");
fprintf("\t1  cos(x) -sin(x) 0  \n");
fprintf("\t1  sin(x)  cos(x) 0  \n");
fprintf("\t0     0      0     1\n\n");
fprintf('Now, for translation by a vector "T(x,y,z)":\ntranslation
matrix:\n\n');
fprintf("\t1   0   0   T(x) \n");
fprintf("\t0   1   0   T(y) \n");
fprintf("\t0   0   1   T(z) \n");
fprintf("\t0   0   0    1 \n\n");
fprintf("We now form the matrix A/B (denoted here as 'AB') by
multiplying the rotation and translation matrices, which is: \n");
AB
fprintf("\nSo, the new position (P) in old frame(A) can be
expressed as:\n\t (P in A) = (A/B) * (P in B)\n");
fprintf("Where , P in B :- coordinates of final point in the new
frame, represented as a 4*1 matrix :-\n");
p_in_b=zeros(4,1);
for i = 1:3
    p_in_b(i,1)=final(i);
end
p_in_b(4,1)=1;
p_in_b
fprintf("\nUpon multiplying AB and p_in_b, we get a 4*1 matrix,
and our answer is first three elements\n");
result
fprintf("Out of the given options, Option-D is a matrix that has
these same eigenvalues.\n");

% Functions used while solving:
function rot = get_rot(angle)
    rot=zeros(4,4);
    rot(1,1)=1;
```

```
        rot(4,4)=1;
        rot(2,2)=cos(angle);
        rot(3,3)=cos(angle);
        rot(2,3)=-sin(angle);
        rot(3,2)=sin(angle);
    end
    function trans = get_trans(initial)
        trans=eye(4);
        for i = 1:3
            trans(i,4)=initial(i);
        end
    end
    function AB = get_AB(rot,trans)
        AB=zeros(4,4);
        AB=rot*trans;
    end
    function res = get_result(AB,final)
        next=zeros(4,1);
        next(4,1)=1;
        for i = 1:3
            next(i,1)=final(i);
        end
        AB=AB*next;
        res=zeros(1,3);
        for i = 1:3
            res(i)=AB(i,1);
        end
    end
end



% ================================================   QUESTION - 4
 ================================================
function q4(ques,vari)
    variant(ques,vari);
    n=4;
    A=rand(n,n);
    for i = 1:n
        for j = 1:n
            A(i,j)=A(i,j)-5+randi(10);
        end
    end
    [W,S,V_dash]=svd(A);
    V=V_dash';
    [L,U,f]=lu(V);
    fprintf("You are given a matrix");
    A
    fprintf("The singular value decomposition of A is given by A = W S
 V'.\n");
    fprintf("The LU Decomposition of V is represented as V = LU ,
 where : \n");
    fprintf("\t L := Lower triangular matrix\n\t U := Upper triangular
 matrix\n\n");
```

```
    fprintf("Which of the following matrices have the eigen-values
 same as the matrix U.\n");
    fprintf("Options:\n\n");
    eigs_of_u=eigs(U);
    Id=["A.","B.","C.","D."];
    for i = 1:4
        now=rand(1,n);
        for j = 1:n
            now(j)=now(j)-0.5+rand(1);
        end
        if i==4
            now=eigs_of_u;
        end
        fprintf("%s \n",Id(i));
        r=rand(n);
        now_diag=diag(now);
        opt=inv(r)*now_diag*r;
        disp(opt);
    end
    fprintf("Answer: D\nExplanation:\n\n");
    fprintf("For the given matrix A, we express the unique singular
 value decomposition (SVD) as:\n");
 fprintf("\n\t A  =  W * S * V_dash\nAs: \n");
 W
 S
 V_dash
    fprintf("\nHere, we have the last matrix as V' (transpose of V).
 So, we revert it back to V.");
    V
    fprintf("We now express V as the linear decomposition of 2
 matrices L,U - lower, and upper triangular matrices respectively.
\n");
    fprintf("\n\t V = L * U\nAs\n");
 L
 U
    fprintf("\nThe eigen values of the matrix U are : ");
    eigs_of_u
    fprintf("Out of the given matrices, Option D. has the same eigen
 values as that of U.\n");
end


% ===============================================   QUESTION - 5
 ===============================================
function q5(ques,vari)
    variant(ques,vari);
    C= rand(1,1);
    D= rand(1,1);
    e= rand(1,1);
    f= rand(1,1);
    g= rand(1,1);
    h= rand(1,1);
```

```
    fprintf( "The energy of a particle in the 2D coordinate system is
 defined as \n\t E = %.2f*(x(2) - x(1)^2)^2 + (%.2f - x(1))^2 Joules.
\n\n",C,D);
    fprintf( "It is defined in the region such that: \n    x(1) +
 %.2f*x(2) <= %.2f \n    %.2f*x(1) + x(2) = %.2f \n    x(1) >= 0, x(2)
 >= 0\n\n",e,f,g,h);
    fprintf( "Find the position of minimum energy of the particle.
\n");
    fun = @(x) C*(x(2)-x(1)^2)^2 + (D-x(1))^2;

    % constraints are written in the below form
        %c(x) <=0
        %ceq(x) = 0
        %A.x <=b
        %Aeq.x = beq
        %lb <= x <= ub

    x0 = [0.5,0]; % initial guess
    A = [1,e];
    b = f;
    Aeq = [g,1];
    beq = h;
    lb = [0,0];
    x = fmincon(fun,x0,A,b,Aeq,beq);

    fprintf("\nOptions\n");
    Id=["A.","B.","C.","D."];

    options_generation = [1+rand(1,1) rand(1,1) rand(1,1) rand(1,1)
 rand(1,1) rand(1,1)];
    for i = 1:3
        fprintf("%s  %.4f %.4f \n",Id(i), options_generation(i),
 options_generation(i+3));
    end
        fprintf("%s  %.4f %.4f \n",Id(4), x)

    fprintf("\nAnswer : D\n");

    fprintf("\n Explanation:\n");
    fprintf(" The energy of particle in 2D coordinate system is given
 as \n\t E = %.2f*(q - x(1)^2)^2 + (%.2f - x(1))^2 Joules.\n\n",C,D);
    fprintf(" After writing constraints in the form: \n")
    fprintf(" c(x) <=0 \n")
    fprintf(" ceq(x) = 0\n")
    fprintf(" A.x <=b \n")
    fprintf(" Aeq.x = beq \n")
    fprintf(" lb <= x <= ub \n")
    fprintf(" \n we get: \n")
    x0
    A
    b
    Aeq
    beq
    lb
```

```
    fprintf(" Solving this using the function 'fmincon' gives \n")
    x
end



% ===============================================   QUESTION - 6
 ===============================================
function q6(ques,vari)
    variant(ques,vari);
    a= randi([1,6],1,1);
    b= randi([1,5],1,1);
    step_size = 0.01* randi([1,7],1,1);
    iterations = 50*randi([2,6],1,1);
    fprintf( "You are stuck at a high altitude on a snow mountain
 while skewing.\n" )
    fprintf( "There is a drone, which can capture the topography of
 the region around you and converts it into a mathematical expression:
\n\n");
    fprintf("\tf(x,y) = (x - %.2f)^2 + (%.2f - xy)^2.\n\n",a,b);
    fprintf( "You have to reach the possible lowest altitude by
 following a steepest decent path at every step with the step size of
 %0.2f and find the lowest possible altitude w.r.t mean sea level that
 can be reached in %d steps.\n",step_size,iterations)
    fprintf("Starting guess can be taken as [2,1]\n");

    %f= @(x) x(1).^2+ (x(2)-1).^2
    syms x y ;
    %f= @(x,y) x^2+ (x*y-4)^2
    f= (x-a)^2 + (b-x*y)^2;
    %gradf = @(x,y) [ 2*x(1), 2*x(2)-2];
    dfdx=diff(f,x);
    dfdy=diff(f,y);

    %starting_guess = [2*rand(1,1),rand(1,1)]
    starting_guess = [2,1];
    gradf = [vpa(subs(dfdx, {x,y}, starting_guess)) subs(dfdy, {x,y},
 starting_guess)];
    %epsilon = 0.0000001;

    guesses = [starting_guess];
    next_guess = starting_guess;
    optimum_value = subs(f, {x,y}, gradf) ;
    k(iterations,2)=0;

    for i=1:iterations
        next_guess = next_guess - step_size*gradf;
        k(i,:) =  next_guess;
        gradf = [round(subs(dfdx, {x,y}, next_guess),3)
 round(subs(dfdy, {x,y}, next_guess),3)];
        if optimum_value > subs(f, {x,y}, next_guess)
            optimum_value = subs(f, {x,y}, next_guess);
            optimal_point = next_guess;
        end
```

```
    end

    optimum_value = round(vpa(optimum_value),3);
    optimal_point;

    fprintf("\nOptions\n");
    Id=["A.","B.","C.","D."];
    for i = 1:3
        fprintf("%s ",Id(i));
        var=-1+randi(2)+rand(1);
        if var == optimum_value
            var=var+rand(1);
        end
        fprintf("%f  \n",var);
    end
    fprintf("%s  %f \n",Id(4), optimum_value);
    fprintf("Answer: D\n");
    fprintf("\nExplanation:\n");
    fprintf("Partial derivative of 'f' at (x,y) w.r.t x & y is
 calculated as: \n")
    dfdx=diff(f,x)
    dfdy=diff(f,y)
    fprintf(" Starting guess is taken as [2,1]\n");
    fprintf(" For each iteration a new (x,y) is calculated using
 the step size And the formula [x(i+1),y(i+1)]=[x(i),y(i)]-
 step_size*grad(f)@(x(i),y(i)).\n")
    fprintf(" After the given number of iterations, 'optimum_value'
 and 'optimum_point' is calculated from the value of function at each
 step \n")
    optimum_value
    optimal_point
end
```

CH5019 - Project
Group no. - 25

Q 1. - Type: Numerical
<Q. 1, V. 1>

In a city X, there is a 24/7 vaccination center where the arrival of people follows a poison distribution.Assume a typical demographic distribution of

Number	Age Group
26 :-	0-18
27 :-	18-45
29 :-	45-60
18 :-	above 60

The arrival rate of each age group at the centre is proportional to their population percentage, and the sum of arrival rates is proportional to 15.

Given that vaccines are delivered at the rate of 13/hour , find the probability that in the duration of 1 hour, the more people from the older population get vaccinated than the younger ones.

Options:
A. 0.102
B. 0.002
C. 0.357
D. 0.157

Answer:B
Explanation:

Probablity of k possibilities in a time limit of x with a poisson ratio of lambda is
probability =

*(exp(-lambda*x)*(lambda*x)^k)/factorial(k)*

For a total of 13 vaccines per hour the possible permutations are:- (in the descending order of age groups)
permutations =

7	3	2	1
6	4	2	1
5	4	3	1

The total probability is the sum of all these possibilities
= 0.002

<Q. 1, V. 2>

In a city X, there is a 24/7 vaccination center where the arrival of people follows a poison distribution.Assume a typical demographic distribution of

Number	Age Group
27 :-	0-18
32 :-	18-45
22 :-	45-60
19 :-	above 60

The arrival rate of each age group at the centre is proportional to their population percentage, and the sum of arrival rates is proportional to 12.
Given that vaccines are delivered at the rate of 13/hour , find the probability that in the duration of 1 hour, the more people from the older population get vaccinated than the younger ones.

Options:
A. 0.357
B. 0.002

C. 0.157
D. 0.102

Answer:B
Explanation:

Probablity of k possibilities in a time limit of x with a poisson ratio of lambda is
probability =

*(exp(-lambda*x)*(lambda*x)^k)/factorial(k)*

For a total of 13 vaccines per hour the possible permutations are:- (in the descending order of age groups)
permutations =

7	3	2	1
6	4	2	1
5	4	3	1

The total probability is the sum of all these possibilities
= 0.002

--

<Q. 1, V. 3>

In a city X, there is a 24/7 vaccination center where the arrival of people follows a poison distribution.Assume a typical demographic distribution of

Number	Age Group
26 :-	0-18
27 :-	18-45
29 :-	45-60
18 :-	above 60

The arrival rate of each age group at the centre is proportional to their population percentage, and the sum of arrival rates is proportional to 11.
Given that vaccines are delivered at the rate of 13/hour , find the probability that in the duration of 1 hour, the more people from the older population get vaccinated than the younger ones.

Options:
A. 0.157
B. 0.357
C. 0.002
D. 0.102

Answer:C
Explanation:

Probablity of k possibilities in a time limit of x with a poisson ratio of lambda is
probability =

*(exp(-lambda*x)*(lambda*x)^k)/factorial(k)*

For a total of 13 vaccines per hour the possible permutations are:- (in the descending order of age groups)
permutations =

7	3	2	1
6	4	2	1
5	4	3	1

The total probability is the sum of all these possibilities
= 0.002

<Q. 1, V. 4>

In a city X, there is a 24/7 vaccination center where the arrival of people follows a poison distribution.Assume a typical demographic distribution of

Number Age Group
26 :- 0-18
27 :- 18-45
29 :- 45-60
18 :- above 60

The arrival rate of each age group at the centre is proportional to their population percentage, and the sum of arrival rates is proportional to 15.
Given that vaccines are delivered at the rate of 13/hour , find the probability that in the duration of 1 hour, the more people from the older population get vaccinated than the younger ones.

Options:
A. 0.157
B. 0.002
C. 0.357
D. 0.102

Answer:B
Explanation:

Probablity of k possibilities in a time limit of x with a poisson ratio of lambda is
probability =

*(exp(-lambda*x)*(lambda*x)^k)/factorial(k)*

For a total of 13 vaccines per hour the possible permutations are:- (in the descending order of age groups)
permutations =

7	3	2	1
6	4	2	1
5	4	3	1

The total probability is the sum of all these possibilities
= 0.002

<Q. 1, V. 5>

In a city X, there is a 24/7 vaccination center where the arrival of people follows a poison distribution.Assume a typical demographic distribution of

Number Age Group
24 :- 0-18
30 :- 18-45
32 :- 45-60
14 :- above 60

The arrival rate of each age group at the centre is proportional to their population percentage, and the sum of arrival rates is proportional to 14.
Given that vaccines are delivered at the rate of 13/hour , find the probability that in the duration of 1 hour, the more people from the older population get vaccinated than the younger ones.

Options:
A. 0.102
B. 0.157
C. 0.357
D. 0.002

Answer:D
Explanation:

Probablity of k possibilities in a time limit of x with a poisson ratio of lambda is
probability =

*(exp(-lambda*x)*(lambda*x)^k)/factorial(k)*

For a total of 13 vaccines per hour the possible permutations are:- (in the descending order of age groups)
permutations =

7	3	2	1
6	4	2	1
5	4	3	1

The total probability is the sum of all these possibilities
= 0.002

--

Q 2. - Type: Numerical
<Q. 2, V. 1>

You are the regional manager of a famous paper selling company, and make sales over a period of 15 days. The profits for each day automatically gets parsed to a specific software. The corporate expects you to have made profits in an increasing order.Unfortunately, the sort function in that software is faulty and doesn't always yield the right answer.
The sort function in this software performs sequential operations of the type [D_i ,P_i], which means that the profits-list from indices (days) [1,D_i] would be sorted with a probability of P_i , or would remain the same with a probability of 1-P_i .

Given, profits during the given period: [439 382 187 32 695 35 317 277 47 98 706 765 795 823 950]

Find the probability that the profits' list would be sorted after performing ALL of the below operations.
Sequential operations: (of the form [D_i , P_i])
[1 , 0.3410]
[2 , 0.4150]
[3 , 0.2240]
[4 , 0.2490]
[5 , 0.2550]
[6 , 0.4940]
[7 , 0.3010]
[8 , 0.1090]
[9 , 0.0410]
[10 , 0.4530]
[11 , 0.1390]
[12 , 0.1500]
[13 , 0.2580]
[14 , 0.1600]
[15 , 0.2550]

Options
A. 0.5370
B. 0.3800
C. 0.0510
D. 0.8141

Answer: D
Explanation:
Firstly, we make the actual sorted profits array and compare it with the given array.

Consider 'idx' as the largest index such that profits[idx] != sorted_profits[idx] holds. (which in this case is 10).

*So, we are not interested in the operations with D_i less than idx, since the array will still be unsorted. Now, let us look at the case where we *never* get a sorted array. The probability for that to happen is product of all (1-P_i)'s for every i>=idx .*

The final answer is 1 - (the above result) , that is, 1 - (product of all (1-P_i)'s') for every i >= idx
So, the probability will be :

= 1 - (1 - 0.453000)(1 - 0.139000)*(1 - 0.150000)*(1 - 0.258000)*(1 - 0.160000)*(1 - 0.255000)*
= 0.8141

<Q. 2, V. 2>

You are the regional manager of a famous paper selling company, and make sales over a period of 15 days. The profits for each day automatically gets parsed to a specific software. The corporate expects you to have made profits in an increasing order.Unfortunately, the sort function in that software is faulty and doesn't always yield the right answer.
The sort function in this software performs sequential operations of the type [D_i ,P_i], which means that the profits-list from indices (days) [1,D_i] would be sorted with a probability of P_i , or would remain the same with a probability of 1-P_i .

Given, profits during the given period: [212 558 78 167 185 317 251 556 218 314 622 703 707 893 913]

Find the probability that the profits' list would be sorted after performing ALL of the below operations.
Sequential operations: (of the form [D_i , P_i])
[1 , 0.3810]
[2 , 0.1610]
[3 , 0.2420]
[4 , 0.1290]
[5 , 0.3510]
[6 , 0.3150]
[7 , 0.2940]
[8 , 0.4690]
[9 , 0.1680]
[10 , 0.4030]
[11 , 0.3350]
[12 , 0.2990]
[13 , 0.4530]
[14 , 0.4230]
[15 , 0.3600]

Options

A. 0.6560
B. 0.5910
C. 0.9100
D. 0.9438

Answer: D
Explanation:
Firstly, we make the actual sorted profits array and compare it with the given array.

Consider 'idx' as the largest index such that profits[idx] != sorted_profits[idx] holds. (which in this case is 10).

*So, we are not interested in the operations with D_i less than idx, since the array will still be unsorted. Now, let us look at the case where we *never* get a sorted array. The probability for that to happen is product of all (1-P_i)'s for every i>=idx .*

The final answer is 1 - (the above result) , that is, 1 - (product of all (1-P_i)'s') for every i >= idx
So, the probability will be :

= 1 - (1 - 0.403000)(1 - 0.335000)*(1 - 0.299000)*(1 - 0.453000)*(1 - 0.423000)*(1 - 0.360000)*
= 0.9438

<Q. 2, V. 3>

You are the regional manager of a famous paper selling company, and make sales over a period of 15 days. The profits for each day automatically gets parsed to a specific software. The corporate expects you to have made profits in an increasing order.Unfortunately, the sort function in that software is faulty and doesn't always yield the right answer.
The sort function in this software performs sequential operations of the type [D_i ,P_i], which means that the profits-list from indices (days) [1,D_i] would be sorted with a probability of P_i , or would remain the same with a probability of 1-P_i .

Given, profits during the given period: [408 179 633 24 624 111 233 548 328 127 607 803 884 980 999]

Find the probability that the profits' list would be sorted after performing ALL of the below operations.
Sequential operations: (of the form [D_i , P_i])
[1 , 0.3150]
[2 , 0.0910]
[3 , 0.3890]
[4 , 0.1000]
[5 , 0.1940]
[6 , 0.2460]
[7 , 0.3460]

```
[  8 , 0.4190 ]
[  9 , 0.1560 ]
[ 10 , 0.1810 ]
[ 11 , 0.3750 ]
[ 12 , 0.2620 ]
[ 13 , 0.1950 ]
[ 14 , 0.0680 ]
[ 15 , 0.0500 ]
```

Options
A. 0.0630
B. 0.4580
C. 0.6020
D. 0.6713

Answer: D
Explanation:
Firstly, we make the actual sorted profits array and compare it with the given array.

Consider 'idx' as the largest index such that profits[idx] != sorted_profits[idx] holds. (which in this case is 11).

*So, we are not interested in the operations with D_i less than idx, since the array will still be unsorted. Now, let us look at the case where we *never* get a sorted array. The probability for that to happen is product of all (1-P_i)'s for every i>=idx .*

The final answer is 1 - (the above result) , that is, 1 - (product of all (1-P_i)'s') for every i >= idx
So, the probability will be :

$$= 1 - (1 - 0.375000)*(1 - 0.262000)*(1 - 0.195000)*(1 - 0.068000)*(1 - 0.050000)$$
$$= 0.6713$$

<Q. 2, V. 4>

You are the regional manager of a famous paper selling company, and make sales over a period of 15 days. The profits for each day automatically gets parsed to a specific software. The corporate expects you to have made profits in an increasing order.Unfortunately, the sort function in that software is faulty and doesn't always yield the right answer.
The sort function in this software performs sequential operations of the type [D_i ,P_i], which means that the profits-list from indices (days) [1,D_i] would be sorted with a probability of P_i , or would remain the same with a probability of 1-P_i .

Given, profits during the given period: [85 141 433 242 593 648 585 96 140 353 650 668 752 857 879]

```
Find the probability that the profits' list would be sorted after
 performing ALL of the below operations.
Sequential operations: (of the form [D_i , P_i])
[  1 , 0.2010 ]
[  2 , 0.0500 ]
[  3 , 0.2830 ]
[  4 , 0.3470 ]
[  5 , 0.4900 ]
[  6 , 0.0280 ]
[  7 , 0.2520 ]
[  8 , 0.4320 ]
[  9 , 0.2990 ]
[ 10 , 0.2560 ]
[ 11 , 0.1140 ]
[ 12 , 0.4470 ]
[ 13 , 0.1840 ]
[ 14 , 0.0990 ]
[ 15 , 0.1410 ]

Options
A.  0.2520
B.  0.5080
C.  0.2680
D.  0.7698

Answer: D
Explanation:
Firstly, we make the actual sorted profits array and compare it with
 the given array.

 Consider 'idx' as the largest index such that profits[idx] !=
 sorted_profits[idx] holds. (which in this case is 10).

So, we are not interested in the operations with D_i less than idx,
 since the array will still be unsorted. Now, let us look at the case
 where we *never* get a sorted array. The probability for that to
 happen is product of all (1-P_i)'s for every i>=idx .

The final answer is 1 - (the above result) , that is, 1 - (product of
 all (1-P_i)'s') for every i >= idx
So, the probability will be :

= 1 - (1 - 0.256000)*(1 - 0.114000)*(1 - 0.447000)*(1 - 0.184000)*(1 -
 0.099000)*(1 - 0.141000)
= 0.7698

--------------------------------------------------------------------------------

<Q. 2, V. 5>

You are the regional manager of a famous paper selling company,
 and make sales over a period of 15 days. The profits for each
 day automatically gets parsed to a specific software. The
 corporate expects you to have made profits in an increasing
```

order.Unfortunately, the sort function in that software is faulty and doesn't always yield the right answer.
The sort function in this software performs sequential operations of the type [D_i ,P_i], which means that the profits-list from indices (days) [1,D_i] would be sorted with a probability of P_i , or would remain the same with a probability of 1-P_i .

Given, profits during the given period: [446 729 530 548 134 509 745 717 106 167 749 842 859 903 984]

Find the probability that the profits' list would be sorted after performing ALL of the below operations.
Sequential operations: (of the form [D_i , P_i])
[1 , 0.3260]
[2 , 0.1200]
[3 , 0.1340]
[4 , 0.1030]
[5 , 0.0410]
[6 , 0.1530]
[7 , 0.1530]
[8 , 0.1560]
[9 , 0.0900]
[10 , 0.4540]
[11 , 0.3310]
[12 , 0.1690]
[13 , 0.2100]
[14 , 0.2880]
[15 , 0.4730]

Options
A. 0.7810
B. 0.1470
C. 0.4270
D. 0.9100

Answer: D
Explanation:
Firstly, we make the actual sorted profits array and compare it with the given array.

Consider 'idx' as the largest index such that profits[idx] != sorted_profits[idx] holds. (which in this case is 10).

*So, we are not interested in the operations with D_i less than idx, since the array will still be unsorted. Now, let us look at the case where we *never* get a sorted array. The probability for that to happen is product of all (1-P_i)'s for every i>=idx .*

The final answer is 1 - (the above result) , that is, 1 - (product of all (1-P_i)'s') for every i >= idx
So, the probability will be :

= 1 - (1 - 0.454000)(1 - 0.331000)*(1 - 0.169000)*(1 - 0.210000)*(1 - 0.288000)*(1 - 0.473000)*

= 0.9100

--

Q 3. - Type: Numerical
<Q. 3, V. 1>

A housefly in a room, travels to the point (1, 8, -8), and disorients its path by an angle of 90 degrees with respect to the positive X-axis, and finally reaches the point P(-1, 9, 3) with respect to this new frame.
Let this final coordinate, when read by the initial coordinate frame, be Q(x,y,z). Which of the following matrices has eigenvalues equal to the coordinates of Q?
(Consider the starting point of the housefly as the origin in the initial frame)

Options:
A.

-1.0787	-1.5769	-5.7676
6.7253	6.3787	6.9907
1.1381	0.7038	5.9332

B.

-0.1423	-0.3094	-0.4136
56.3911	3.1437	0.2394
-39.1547	-0.1223	2.7979

C.

0.8244	0.0663	1.9247
3.2941	-0.7387	4.8859
-1.4336	-0.0858	-3.0433

D.

-333.3817	-93.2992	-413.2484
5.4532	-10.1351	9.9483
296.8534	92.0453	365.5167

Answer: D.
Explanation:

Let us name the initial frame as A, and the final frame (after translation and rotation) as B.
Firstly we formulate the rotation matrix.
It is given as: (for an angle 'x' , wrt Positive X-axis)

1	0	0	0
1	cos(x)	-sin(x)	0
1	sin(x)	cos(x)	0
0	0	0	1

Now, for translation by a vector "T(x,y,z)":

translation matrix:

```
 1   0   0   T(x)
 0   1   0   T(y)
 0   0   1   T(z)
 0   0   0    1
```

We now form the matrix A/B (denoted here as 'AB') by multiplying the rotation and translation matrices, which is:

AB =

```
    1.0000         0         0    1.0000
         0    0.0000   -1.0000    8.0000
         0    1.0000    0.0000    8.0000
         0         0         0    1.0000
```

So, the new position (P) in old frame(A) can be expressed as:
*(P in A) = (A/B) * (P in B)*
*Where , P in B :- coordinates of final point in the new frame, represented as a 4*1 matrix :-*

p_in_b =

```
 -333.3817
    5.4532
  296.8534
    1.0000
```

*Upon multiplying AB and p_in_b, we get a 4*1 matrix, and our answer is first three elements*

result =

```
     0     5    17
```

Out of the given options, Option-D is a matrix that has these same eigenvalues.

<Q. 3, V. 2>

A housefly in a room, travels to the point (3, -6, -7), and disorients its path by an angle of 105 degrees with respect to the positive X-axis, and finally reaches the point P(-3, -4, 2) with respect to this new frame.
Let this final coordinate, when read by the initial coordinate frame, be Q(x,y,z). Which of the following matrices has eigenvalues equal to the coordinates of Q?
(Consider the starting point of the housefly as the origin in the initial frame)

Options:

A.

```
   -3.3072   -0.5887   -0.7832
   -3.1326   -3.8246   -1.1865
    1.9455    1.0765   -1.0782
```

B.

```
   1.0e+03 *

    0.2047    0.1029    0.2766
    2.0326    1.0502    2.8108
   -0.9048   -0.4649   -1.2454
```

C.

```
    4.3748    4.2251    4.3628
    1.8291    1.4734    2.2165
   -3.6464   -3.4683   -4.4046
```

D.

```
  -45.0420  -72.7096  -20.9251
   14.5551   26.7420    6.3971
   36.9836   58.1777   17.3527
```

Answer: D.

Explanation:

Let us name the initial frame as A, and the final frame (after translation and rotation) as B.

Firstly we formulate the rotation matrix.

It is given as: (for an angle 'x' , wrt Positive X-axis)

```
 1     0      0     0
 1  cos(x) -sin(x) 0
 1  sin(x)  cos(x) 0
 0     0      0     1
```

Now, for translation by a vector "T(x,y,z)":

translation matrix:

```
 1   0   0   T(x)
 0   1   0   T(y)
 0   0   1   T(z)
 0   0   0    1
```

We now form the matrix A/B (denoted here as 'AB') by multiplying the rotation and translation matrices, which is:

AB =

```
    1.0000         0         0    3.0000
         0   -0.2588   -0.9659    8.3144
```

```
         0    0.9659   -0.2588   -3.9838
         0         0         0    1.0000
```

So, the new position (P) in old frame(A) can be expressed as:
(P in A) = (A/B) * (P in B)
Where , P in B :- coordinates of final point in the new frame, represented as a 4*1 matrix :-

p_in_b =

```
  -45.0420
   14.5551
   36.9836
    1.0000
```

Upon multiplying AB and p_in_b, we get a 4*1 matrix, and our answer is first three elements

result =

```
         0    7.4178   -8.3652
```

Out of the given options, Option-D is a matrix that has these same eigenvalues.

<Q. 3, V. 3>

A housefly in a room, travels to the point (6, -5, -9), and disorients its path by an angle of 75 degrees with respect to the positive X-axis, and finally reaches the point P(9, -4, 2) with respect to this new frame.
Let this final coordinate, when read by the initial coordinate frame, be Q(x,y,z). Which of the following matrices has eigenvalues equal to the coordinates of Q?
(Consider the starting point of the housefly as the origin in the initial frame)

Options:
A.

```
    0.9718   -1.1961   -0.5767
  -13.9359  -19.4004  -10.6383
   22.2728   31.8254   17.3283
```

B.

```
    3.4029   -0.5547   -0.8714
    0.6084    5.0447    0.3032
    1.2305    1.0978    6.3228
```

C.

```
   -0.9454   -0.9743   -1.1525
    5.0768    2.9903    5.7693
   -4.5295   -2.6983   -5.1865
```

D.

```
  -21.3442  -11.5895   -1.0967
   36.5171   27.4726    2.1347
  -26.7743  -17.6717    2.7987
```

Answer: D.
Explanation:

Let us name the initial frame as A, and the final frame (after translation and rotation) as B.
Firstly we formulate the rotation matrix.
It is given as: (for an angle 'x' , wrt Positive X-axis)

```
 1     0      0     0
 1  cos(x) -sin(x) 0
 1  sin(x)  cos(x) 0
 0     0      0     1
```

Now, for translation by a vector "T(x,y,z)":
translation matrix:

```
 1   0   0   T(x)
 0   1   0   T(y)
 0   0   1   T(z)
 0   0   0    1
```

We now form the matrix A/B (denoted here as 'AB') by multiplying the rotation and translation matrices, which is:

AB =

```
    1.0000         0         0    6.0000
         0    0.2588   -0.9659    7.3992
         0    0.9659    0.2588   -7.1590
         0         0         0    1.0000
```

So, the new position (P) in old frame(A) can be expressed as:
*(P in A) = (A/B) * (P in B)*
*Where , P in B :- coordinates of final point in the new frame, represented as a 4*1 matrix :-*

p_in_b =

```
  -21.3442
   36.5171
  -26.7743
    1.0000
```

*Upon multiplying AB and p_in_b, we get a 4*1 matrix, and our answer is first three elements*

result =

```
   15.0000    4.4321  -10.5051
```

Out of the given options, Option-D is a matrix that has these same eigenvalues.

<Q. 3, V. 4>

A housefly in a room, travels to the point (-8, -7, 0), and disorients its path by an angle of 120 degrees with respect to the positive X-axis, and finally reaches the point P(8, 0, -9) with respect to this new frame.
Let this final coordinate, when read by the initial coordinate frame, be Q(x,y,z). Which of the following matrices has eigenvalues equal to the coordinates of Q?
(Consider the starting point of the housefly as the origin in the initial frame)

Options:
A.

```
   -3.2841   -3.9911   -3.0550
    4.5893    6.3886    3.2890
   -0.3203   -0.8699    0.6448
```

B.

```
   -1.1313   -2.4639   -2.8912
    4.3848    6.6922    7.5245
   -2.4341   -3.1566   -3.4440
```

C.

```
   -4.3597   -4.5778   -1.5746
    1.7640    1.8945    1.7992
    0.0782    0.3768   -2.9063
```

D.

```
   -0.7217  -22.8802  -28.4969
   -0.1223    0.7223    2.8892
    0.4964    9.8562    9.7314
```

Answer: D.
Explanation:

Let us name the initial frame as A, and the final frame (after translation and rotation) as B.
Firstly we formulate the rotation matrix.

```
It is given as: (for an angle 'x' , wrt Positive X-axis)

 1     0     0     0
 1  cos(x) -sin(x) 0
 1  sin(x)  cos(x) 0
 0     0     0     1

Now, for translation by a vector "T(x,y,z)":
translation matrix:

 1   0   0   T(x)
 0   1   0   T(y)
 0   0   1   T(z)
 0   0   0    1

We now form the matrix A/B (denoted here as 'AB') by multiplying the
 rotation and translation matrices, which is:

AB =

    1.0000         0         0   -8.0000
         0   -0.5000   -0.8660    3.5000
         0    0.8660   -0.5000   -6.0622
         0         0         0    1.0000


So, the new position (P) in old frame(A) can be expressed as:
  (P in A) = (A/B) * (P in B)
Where , P in B :- coordinates of final point in the new frame,
 represented as a 4*1 matrix :-

p_in_b =

   -0.7217
   -0.1223
    0.4964
    1.0000


Upon multiplying AB and p_in_b, we get a 4*1 matrix, and our answer is
 first three elements

result =

         0   11.2942   -1.5622

Out of the given options, Option-D is a matrix that has these same
 eigenvalues.

----------------------------------------------------------------------

<Q. 3, V. 5>
```

A housefly in a room, travels to the point (-1, -5, 0), and disorients its path by an angle of 150 degrees with respect to the positive X-axis, and finally reaches the point P(-4, 2, -9) with respect to this new frame.
Let this final coordinate, when read by the initial coordinate frame, be Q(x,y,z). Which of the following matrices has eigenvalues equal to the coordinates of Q?
(Consider the starting point of the housefly as the origin in the initial frame)

Options:

A.

2.9504	3.8548	4.4336
0.1929	-2.0209	0.1673
-3.9550	-2.9786	-5.5947

B.

-6.3650	-6.2333	-4.0631
3.3752	6.6353	3.1929
2.3163	0.1604	1.6105

C.

-3.4859	-2.9188	-1.7310
9.4534	7.5566	4.6315
-8.0755	-5.9052	-4.1162

D.

7.3317	1.1178	-1.2628
-9.7609	-7.4122	-1.5489
6.2230	9.0213	8.4728

Answer: D.
Explanation:

Let us name the initial frame as A, and the final frame (after translation and rotation) as B.
Firstly we formulate the rotation matrix.
It is given as: (for an angle 'x' , wrt Positive X-axis)

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 1 & \cos(x) & -\sin(x) & 0 \\ 1 & \sin(x) & \cos(x) & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Now, for translation by a vector "T(x,y,z)":
translation matrix:

$$\begin{matrix} 1 & 0 & 0 & T(x) \\ 0 & 1 & 0 & T(y) \\ 0 & 0 & 1 & T(z) \\ 0 & 0 & 0 & 1 \end{matrix}$$

We now form the matrix A/B (denoted here as 'AB') by multiplying the rotation and translation matrices, which is:

```
AB =

    1.0000         0         0   -1.0000
         0   -0.8660   -0.5000    4.3301
         0    0.5000   -0.8660   -2.5000
         0         0         0    1.0000
```

So, the new position (P) in old frame(A) can be expressed as:
*(P in A) = (A/B) * (P in B)*
*Where , P in B :- coordinates of final point in the new frame, represented as a 4*1 matrix :-*

```
p_in_b =

    7.3317
   -9.7609
    6.2230
    1.0000
```

*Upon multiplying AB and p_in_b, we get a 4*1 matrix, and our answer is first three elements*

```
result =

   -5.0000    7.0981    6.2942
```

Out of the given options, Option-D is a matrix that has these same eigenvalues.

Q 4. - Type: Numerical
<Q. 4, V. 1>

You are given a matrix

```
A =

   -1.9034    0.5237    1.7629    0.8398
   -3.1541    3.6503   -3.4243    0.4268
   -3.0906    5.3851    1.6319    5.6316
   -0.9887    4.6493   -1.7218    5.8335
```

The singular value decomposition of A is given by A = W S V'.
The LU Decomposition of V is represented as V = LU , where :
L := Lower triangular matrix
U := Upper triangular matrix

Which of the following matrices have the eigen-values same as the matrix U.

Options:

A.

-1.5158	2.0371	-1.8210	-2.2799
-0.1282	0.9620	0.2057	-0.3362
0.8797	-0.0090	1.6055	1.0006
0.7038	-2.3494	0.0557	1.4271

B.

5.3658	5.8388	4.1543	5.2245
-12.6399	-13.5737	-10.1303	-12.1187
-1.2835	-1.5323	-0.5261	-1.3637
10.6094	11.4429	8.1865	10.1999

C.

-0.5009	-2.9469	-2.5276	-3.0116
0.6805	2.0202	1.5259	1.7368
-0.4260	-0.6455	0.0544	-1.0806
0.5148	1.1568	0.8607	1.9701

D.

-4.8815	-3.8359	-3.2776	-2.8270
2.0124	0.9633	1.4636	1.3560
0.5610	-0.2461	-1.1794	0.1837
5.2035	5.9844	5.4020	3.0993

Answer: D
Explanation:

For the given matrix A, we express the unique singular value decomposition (SVD) as:

*A = W * S * V_dash*
As:

W =

-0.1194	0.2960	0.5686	0.7582
-0.3420	-0.8541	0.3916	-0.0142
-0.6929	0.4236	0.3007	-0.5001
-0.6235	-0.0590	-0.6579	0.4183

S =

11.8742	0	0	0
0	4.8339	0	0
0	0	3.2749	0
0	0	0	0.5665

V_dash =

0.3422	0.1820	-0.7928	-0.4703

```
   -0.6687   -0.1977    0.0879   -0.7113
    0.0761    0.8770    0.3923   -0.2668
   -0.6556    0.3984   -0.4580    0.4491
```

Here, we have the last matrix as V' (transpose of V). So, we revert it back to V.

```
V =

    0.3422   -0.6687    0.0761   -0.6556
    0.1820   -0.1977    0.8770    0.3984
   -0.7928    0.0879    0.3923   -0.4580
   -0.4703   -0.7113   -0.2668    0.4491
```

We now express V as the linear decomposition of 2 matrices L,U - lower, and upper triangular matrices respectively.

```
  V = L * U
As

L =

    1.0000         0         0         0
    0.5932    1.0000         0         0
   -0.2295    0.2325    1.0000         0
   -0.4317    0.8263    0.6076    1.0000


U =

   -0.7928    0.0879    0.3923   -0.4580
         0   -0.7635   -0.4995    0.7208
         0         0    1.0832    0.1257
         0         0         0   -1.5252
```

The eigen values of the matrix U are :

```
eigs_of_u =

   -1.5252
    1.0832
   -0.7928
   -0.7635
```

Out of the given matrices, Option D. has the same eigen values as that of U.

<Q. 4, V. 2>

You are given a matrix
A =

4.6338	-2.8860	-2.0496	-2.1207
-3.9859	-3.5575	1.6943	2.5579
5.4704	3.6595	-0.7932	0.7523
-1.1137	3.2948	-2.4452	3.8949

The singular value decomposition of A is given by A = W S V'.
The LU Decomposition of V is represented as V = LU , where :
L := Lower triangular matrix
U := Upper triangular matrix

Which of the following matrices have the eigen-values same as the matrix U.
Options:

A.

-0.1602	-0.1171	-0.3417	-0.2718
-0.9887	0.7699	-0.1676	-0.3965
1.1621	0.0040	0.9878	0.4288
0.4839	0.0370	0.1324	0.9498

B.

0.7439	0.0938	-0.0638	0.0813
-0.3857	0.0986	-1.6345	-1.7086
0.3841	0.5061	3.2497	2.9542
-0.3849	-0.4786	-2.1289	-1.8978

C.

-0.5443	-0.6529	-1.5709	-1.7250
0.7408	0.8494	1.3218	1.4804
0.5044	0.3765	1.2355	0.9839
-0.1467	-0.0708	-0.2611	0.0187

D.

-2.6814	-0.9330	-0.5450	-0.7262
6.4522	2.3510	1.3754	2.1560
-3.6057	-1.7720	-1.4883	-1.0945
3.0273	1.3073	-0.5094	-0.7291

Answer: D
Explanation:

For the given matrix A, we express the unique singular value decomposition (SVD) as:

*A = W * S * V_dash*
As:

W =

-0.4237	0.5879	-0.6647	-0.1817
0.6192	0.0785	-0.4913	0.6075
-0.6611	-0.2931	-0.0265	0.6902
-0.0090	-0.7498	-0.5623	-0.3486

```
S =

    9.2861         0         0         0
         0    7.1890         0         0
         0         0    3.3610         0
         0         0         0    2.2367


V_dash =

   -0.8655    0.2286   -0.1907    0.4028
   -0.3693   -0.7677    0.5107   -0.1160
    0.2653    0.1383    0.5730    0.7630
    0.2100   -0.5824   -0.6120    0.4921
```

Here, we have the last matrix as V' (transpose of V). So, we revert it back to V.

```
V =

   -0.8655   -0.3693    0.2653    0.2100
    0.2286   -0.7677    0.1383   -0.5824
   -0.1907    0.5107    0.5730   -0.6120
    0.4028   -0.1160    0.7630    0.4921
```

We now express V as the linear decomposition of 2 matrices L,U - lower, and upper triangular matrices respectively.

```
  V = L * U
As

L =

    1.0000         0         0         0
   -0.2641    1.0000         0         0
   -0.4653    0.3327    1.0000         0
    0.2203   -0.6842    0.8042    1.0000


U =

   -0.8655   -0.3693    0.2653    0.2100
         0   -0.8652    0.2083   -0.5270
         0         0    0.8171    0.7651
         0         0         0   -1.6341
```

The eigen values of the matrix U are :

```
eigs_of_u =

   -1.6341
   -0.8655
   -0.8652
```

```
    0.8171

Out of the given matrices, Option D. has the same eigen values as that
 of U.

-----------------------------------------------------------------------------------

&lt;Q. 4, V. 3&gt;

You are given a matrix
A =

   -2.6329   -3.6129    1.2700    4.1822
    4.2396   -3.5790    3.8440    5.0654
    0.3461   -3.3599   -1.2595    3.6104
   -3.7504    0.7876    1.8261    5.7016

The singular value decomposition of A is given by A = W S V'.
The LU Decomposition of V is represented as V = LU , where :
  L := Lower triangular matrix
  U := Upper triangular matrix

Which of the following matrices have the eigen-values same as the
 matrix U.
Options:

A.
    3.1250    1.3759    2.9226    0.8869
   -1.8755   -0.3418   -2.3026   -0.6235
   -1.5878   -0.8195   -1.4138   -0.4587
   -0.0795   -0.1956    0.2235    0.3654

B.
    0.9287    0.1891   -0.1609    0.0051
    0.5649    0.7874    0.6396    0.2736
   -0.5287   -0.2097    0.1720   -0.2097
   -0.5781   -0.1593   -0.4776    0.4106

C.
   -0.1811   -1.9789   -1.1976   -0.7050
   -0.9005   -1.1262   -1.2572   -0.6954
    1.6137    3.0268    2.7819    1.3476
    0.5628    1.1010    0.8710    0.7993

D.
   -1.8686   -1.6699   -0.4186   -1.0314
    1.0195    1.5435    0.5165    1.5556
   -0.2815   -0.7773   -1.3325   -0.4219
    0.1959    0.1523    0.3094   -0.8989

Answer: D
Explanation:
```

For the given matrix A, we express the unique singular value decomposition (SVD) as:

```
  A  =  W * S * V_dash
As:

W =

   -0.5049    0.2804   -0.4025   -0.7103
   -0.6279   -0.6748    0.3858   -0.0386
   -0.3741   -0.0781   -0.6832    0.6223
   -0.4591    0.6782    0.4716    0.3268


S =

   11.0550         0         0         0
         0    6.7456         0         0
         0         0    3.9350         0
         0         0         0    1.9145


V_dash =

    0.0235   -0.9146    0.1754    0.3636
    0.4493    0.3259    0.6963    0.4550
   -0.3096   -0.1336    0.6845   -0.6463
   -0.8377    0.1985    0.1254    0.4930
```

Here, we have the last matrix as V' (transpose of V). So, we revert it back to V.

```
V =

    0.0235    0.4493   -0.3096   -0.8377
   -0.9146    0.3259   -0.1336    0.1985
    0.1754    0.6963    0.6845    0.1254
    0.3636    0.4550   -0.6463    0.4930
```

We now express V as the linear decomposition of 2 matrices L,U - lower, and upper triangular matrices respectively.

```
  V = L * U
As

L =

    1.0000         0         0         0
   -0.1918    1.0000         0         0
   -0.3975    0.7703    1.0000         0
   -0.0257    0.6031    0.5886    1.0000


U =
```

```
   -0.9146    0.3259   -0.1336    0.1985
         0    0.7589    0.6589    0.1635
         0         0   -1.2070    0.4460
         0         0         0   -1.1937
```

The eigen values of the matrix U are :
eigs_of_u =

```
   -1.2070
   -1.1937
   -0.9146
    0.7589
```

Out of the given matrices, Option D. has the same eigen values as that of U.

<Q. 4, V. 4>

You are given a matrix
A =

```
    3.9726   -0.1016    0.7799   -2.5238
    3.6053    1.8507    2.7014    3.9949
   -1.6618    4.2568    5.4925   -1.5094
    3.9280   -0.7145   -1.0323   -2.4965
```

The singular value decomposition of A is given by A = W S V'.
The LU Decomposition of V is represented as V = LU , where :
L := Lower triangular matrix
U := Upper triangular matrix

Which of the following matrices have the eigen-values same as the matrix U.
Options:

A.

```
    0.8763    0.2258    0.3702    0.4231
    0.2565    0.3704    0.5380   -0.2280
   -0.1113   -0.0586    0.2229   -0.0825
   -0.1250    0.1151   -0.2244    0.6950
```

B.

```
    0.1874    0.1246   -0.0329   -0.0457
   -0.2755   -0.3914   -0.1268   -0.0469
    0.2115    0.2016    0.3930    0.0537
    0.0670    0.0903    0.0289    0.3336
```

C.

```
    1.2052    0.3587    0.3225    0.4399
    0.5647    1.6210    0.4023    1.0727
```

```
   -0.3079   -0.6047    0.5693   -0.7326
   -0.3035   -0.1545   -0.2254    0.5018
```

D.

```
    0.6763   -0.0028   -0.1526    0.1102
    2.6358    1.8696    1.5956    3.1665
    0.3557   -0.0014    1.2606   -0.3605
   -2.0036   -0.7098   -1.2482   -1.6883
```

Answer: D
Explanation:

For the given matrix A, we express the unique singular value decomposition (SVD) as:

```
  A  =  W * S * V_dash
```

As:

```
W =

   -0.0748   -0.6282   -0.3569   -0.6873
    0.3935   -0.5388    0.7421    0.0643
    0.8670   -0.0017   -0.4742    0.1534
   -0.2965   -0.5613   -0.3115    0.7071


S =

    7.8749         0         0         0
         0    6.8537         0         0
         0         0    5.4637         0
         0         0         0    0.4198


V_dash =

   -0.1885   -0.9688    0.1504    0.0567
    0.5890   -0.0787   -0.0707    0.8012
    0.7711   -0.2006   -0.1019   -0.5956
    0.1514    0.1221    0.9808   -0.0128
```

Here, we have the last matrix as V' (transpose of V). So, we revert it back to V.

```
V =

   -0.1885    0.5890    0.7711    0.1514
   -0.9688   -0.0787   -0.2006    0.1221
    0.1504   -0.0707   -0.1019    0.9808
    0.0567    0.8012   -0.5956   -0.0128
```

We now express V as the linear decomposition of 2 matrices L,U - lower, and upper triangular matrices respectively.

```
  V = L * U
As

L =

    1.0000         0         0         0
   -0.0585    1.0000         0         0
    0.1945    0.7586    1.0000         0
   -0.1553   -0.1041   -0.1544    1.0000


U =

   -0.9688   -0.0787   -0.2006    0.1221
         0    0.7966   -0.6073   -0.0057
         0         0    1.2709    0.1320
         0         0         0    1.0196


The eigen values of the matrix U are :
eigs_of_u =

    1.2709
    1.0196
   -0.9688
    0.7966

Out of the given matrices, Option D. has the same eigen values as that
 of U.

----------------------------------------------------------------------------------

<Q. 4, V. 5>

You are given a matrix
A =

    4.7464    1.1199    5.9539    5.4375
    0.7997   -0.4810   -1.0531    0.3208
   -1.0922    0.8220    0.9666   -1.8659
   -2.0254    0.6370    1.0673   -0.8654

The singular value decomposition of A is given by A = W S V'.
The LU Decomposition of V is represented as V = LU , where :
  L := Lower triangular matrix
  U := Upper triangular matrix

Which of the following matrices have the eigen-values same as the
 matrix U.
Options:

A.
    3.2501    2.6490    0.8413    3.2461
    1.4641    2.2508    0.4378    1.8667
```

```
    0.7138    0.7397    1.0502    0.9358
   -3.1616   -3.2939   -1.0719   -3.3364

B.
   -0.7912   -5.2745   -2.5866   -2.6692
   -0.5699   -2.0402   -1.2223   -1.2504
   -0.1398   -0.2947    0.1990   -0.1845
    2.0577    8.8913    4.4007    4.9299

C.
    0.4951   -0.3613   -0.7559   -0.4522
   -0.8467    0.9681    1.5201    0.8043
    2.1932    1.2885    1.9999    0.9408
   -1.5239   -1.2940   -1.9089   -0.6515

D.
   -1.6206   -4.1847   -1.9209   -3.1857
   -0.1490    0.0433    0.5321   -0.0773
   -0.9669    0.0214   -1.3404   -1.4200
    1.2229    2.2091    0.7223    2.4087

Answer: D
Explanation:

For the given matrix A, we express the unique singular value
 decomposition (SVD) as:

  A  =  W * S * V_dash
As:

W =

   -0.9899    0.1254   -0.0272    0.0601
    0.0072   -0.4103   -0.1779    0.8944
    0.1075    0.6193   -0.7667    0.1308
    0.0920    0.6576    0.6163    0.4234


S =

    9.5090         0         0         0
         0    3.4870         0         0
         0         0    0.9906         0
         0         0         0    0.0157


V_dash =

   -0.5255   -0.4994   -0.6888    0.0011
   -0.1015    0.3630   -0.1843    0.9077
   -0.5994    0.7109   -0.0588   -0.3632
   -0.5953   -0.3369    0.6986    0.2100
```

```
Here, we have the last matrix as V' (transpose of V). So, we revert it
 back to V.
V =

   -0.5255   -0.1015   -0.5994   -0.5953
   -0.4994    0.3630    0.7109   -0.3369
   -0.6888   -0.1843   -0.0588    0.6986
    0.0011    0.9077   -0.3632    0.2100

We now express V as the linear decomposition of 2 matrices L,U -
 lower, and upper triangular matrices respectively.

  V = L * U
As

L =

    1.0000         0         0         0
   -0.0016    1.0000         0         0
    0.7250    0.5472    1.0000         0
    0.7628    0.0431   -0.5659    1.0000


U =

   -0.6888   -0.1843   -0.0588    0.6986
         0    0.9074   -0.3633    0.2111
         0         0    0.9523   -0.9588
         0         0         0   -1.6799


The eigen values of the matrix U are :
eigs_of_u =

   -1.6799
    0.9523
    0.9074
   -0.6888

Out of the given matrices, Option D. has the same eigen values as that
 of U.

------------------------------------------------------------------------------------

Q 5. - Type: Numerical
<Q. 5, V. 1>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.15*(x(2) - x(1)^2)^2 + (0.54 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.94*x(2) <= 0.66
   0.39*x(1) + x(2) = 0.26
   x(1) >= 0, x(2) >= 0
```

Find the position of minimum energy of the particle.

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.

Options
A. 1.8479 0.0673
B. 0.9451 0.1816
C. 0.3770 0.5757
D. 0.4997 0.0618

Answer : D

Explanation:
The energy of particle in 2D coordinate system is given as
E = 0.15(q - x(1)^2)^2 + (0.54 - x(1))^2 Joules.*

After writing constraints in the form:
c(x) <=0
ceq(x) = 0
A.x <=b
Aeq.x = beq
lb <= x <= ub

we get:

```
x0 =

    0.5000         0


A =

    1.0000    0.9371


b =

    0.6610


Aeq =

    0.3947    1.0000


beq =
```

```
    0.2590


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.4997    0.0618


-----------------------------------------------------------------------------------

<Q. 5, V. 2>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.19*(x(2) - x(1)^2)^2 + (0.29 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.46*x(2) <= 0.35
   0.32*x(1) + x(2) = 0.46
   x(1) >= 0, x(2) >= 0

Find the position of minimum energy of the particle.

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.


Options
A.  1.2359 0.1588
B.  0.0278 0.8027
C.  0.6585 0.4086
D.  0.1578 0.4097

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.19*(q - x(1)^2)^2 + (0.29 - x(1))^2 Joules.

 After writing constraints in the form:
 c(x) <=0
 ceq(x) = 0
 A.x <=b
```

```
 Aeq.x = beq
 lb <= x <= ub

 we get:

x0 =

    0.5000         0


A =

    1.0000    0.4617


b =

    0.3470


Aeq =

    0.3182    1.0000


beq =

    0.4599


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.1578    0.4097


-----------------------------------------------------------------------------------

<Q. 5, V. 3>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.33*(x(2) - x(1)^2)^2 + (0.75 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.75*x(2) <= 0.17
   0.12*x(1) + x(2) = 0.17
   x(1) >= 0, x(2) >= 0

Find the position of minimum energy of the particle.
```

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.

Options
A. 1.6274 0.1658
B. 0.8419 0.7143
C. 0.5101 0.9070
D. 0.0483 0.1684

Answer : D

Explanation:
The energy of particle in 2D coordinate system is given as
E = 0.33(q - x(1)^2)^2 + (0.75 - x(1))^2 Joules.*

After writing constraints in the form:
c(x) <=0
ceq(x) = 0
A.x <=b
Aeq.x = beq
lb <= x <= ub

we get:

```
x0 =

    0.5000         0


A =

    1.0000    0.7464


b =

    0.1740


Aeq =

    0.1175    1.0000


beq =

    0.1740
```

```
lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.0483    0.1684


--------------------------------------------------------------------------------

<Q. 5, V. 4>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.22*(x(2) - x(1)^2)^2 + (0.87 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.21*x(2) <= 0.84
   0.86*x(1) + x(2) = 0.52
   x(1) >= 0, x(2) >= 0

Find the position of minimum energy of the particle.

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.


Options
A.  1.4774 0.5095
B.  0.8899 0.6208
C.  0.0651 0.7336
D.  0.6522 -0.0371

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.22*(q - x(1)^2)^2 + (0.87 - x(1))^2 Joules.

 After writing constraints in the form:
 c(x) <=0
 ceq(x) = 0
 A.x <=b
 Aeq.x = beq
 lb <= x <= ub
```

```
 we get:

x0 =

    0.5000         0


A =

    1.0000    0.2118


b =

    0.8367


Aeq =

    0.8593    1.0000


beq =

    0.5234


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.6522   -0.0371


-----------------------------------------------------------------------------------

<Q. 5, V. 5>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.23*(x(2) - x(1)^2)^2 + (0.02 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.14*x(2) <= 0.77
   0.97*x(1) + x(2) = 0.39
   x(1) >= 0, x(2) >= 0

Find the position of minimum energy of the particle.

Local minimum found that satisfies the constraints.
```

```
Optimization completed because the objective function is non-
decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint
 tolerance.


Options
A.  1.9934 0.3848
B.  0.3264 0.5626
C.  0.1372 0.6338
D.  0.0976 0.2921

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.23*(q - x(1)^2)^2 + (0.02 - x(1))^2 Joules.

 After writing constraints in the form:
 c(x) <=0
 ceq(x) = 0
 A.x <=b
 Aeq.x = beq
 lb <= x <= ub

 we get:

x0 =

    0.5000         0


A =

    1.0000    0.1390


b =

    0.7695


Aeq =

    0.9698    1.0000


beq =

    0.3868
```

```
lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.0976    0.2921


-------------------------------------------------------------------------------------

Q 6. - Type: Numerical
<Q. 6, V. 1>

You are stuck at a high altitude on a snow mountain while skewing.
There is a drone, which can capture the topography of the region
 around you and converts it into a mathematical expression:

 f(x,y) = (x - 4.00)^2 + (2.00 - xy)^2.

You have to reach the possible lowest altitude by following a steepest
 decent path at every step with the step size of 0.02 and find the
 lowest possible altitude w.r.t mean sea level that can be reached in
 100 steps.
Starting guess can be taken as [2,1]

Options
A. 0.709803
B. 0.113271
C. 1.179986
D.  0.002000
Answer: D

Explanation:
Partial derivative of 'f' at (x,y) w.r.t x & y is calculated as:

dfdx =

2*x + 2*y*(x*y - 2) - 8


dfdy =

2*x*(x*y - 2)

 Starting guess is taken as [2,1]
 For each iteration a new (x,y) is calculated using the
 step size And the formula [x(i+1),y(i+1)]=[x(i),y(i)]-
 step_size*grad(f)@(x(i),y(i)).
 After the given number of iterations, 'optimum_value' and
 'optimum_point' is calculated from the value of function at each
 step
```

optimum_value =

0.002

optimal_point =

[3.96036, 0.50534]

<Q. 6, V. 2>

You are stuck at a high altitude on a snow mountain while skewing. There is a drone, which can capture the topography of the region around you and converts it into a mathematical expression:

f(x,y) = (x - 5.00)^2 + (3.00 - xy)^2.

You have to reach the possible lowest altitude by following a steepest decent path at every step with the step size of 0.04 and find the lowest possible altitude w.r.t mean sea level that can be reached in 150 steps.
Starting guess can be taken as [2,1]

Options
A. 1.526490
B. 1.422041
C. 1.073138
D. 0.000000
Answer: D

Explanation:
Partial derivative of 'f' at (x,y) w.r.t x & y is calculated as:

dfdx =

*2*x + 2*y*(x*y - 3) - 10*

dfdy =

*2*x*(x*y - 3)*

Starting guess is taken as [2,1]
*For each iteration a new (x,y) is calculated using the step size And the formula [x(i+1),y(i+1)]=[x(i),y(i)]-step_size*grad(f)@(x(i),y(i)).*
After the given number of iterations, 'optimum_value' and 'optimum_point' is calculated from the value of function at each step

optimum_value =

0.0

optimal_point =

[4.99972, 0.59992]

--

<Q. 6, V. 3>

You are stuck at a high altitude on a snow mountain while skewing. There is a drone, which can capture the topography of the region around you and converts it into a mathematical expression:

f(x,y) = (x - 4.00)^2 + (5.00 - xy)^2.

You have to reach the possible lowest altitude by following a steepest decent path at every step with the step size of 0.04 and find the lowest possible altitude w.r.t mean sea level that can be reached in 250 steps.
Starting guess can be taken as [2,1]

Options
A. 0.607440
B. 0.799562
C. 1.954087
D. 0.000000
Answer: D

Explanation:
Partial derivative of 'f' at (x,y) w.r.t x & y is calculated as:

dfdx =

*2*x + 2*y*(x*y - 5) - 8*

dfdy =

*2*x*(x*y - 5)*

Starting guess is taken as [2,1]
*For each iteration a new (x,y) is calculated using the step size And the formula [x(i+1),y(i+1)]=[x(i),y(i)]-step_size*grad(f)@(x(i),y(i)).*
After the given number of iterations, 'optimum_value' and 'optimum_point' is calculated from the value of function at each step

optimum_value =

0.0

optimal_point =

[3.99976, 1.25008]

<Q. 6, V. 4>

You are stuck at a high altitude on a snow mountain while skewing. There is a drone, which can capture the topography of the region around you and converts it into a mathematical expression:

f(x,y) = (x - 3.00)^2 + (3.00 - xy)^2.

You have to reach the possible lowest altitude by following a steepest decent path at every step with the step size of 0.05 and find the lowest possible altitude w.r.t mean sea level that can be reached in 300 steps.
Starting guess can be taken as [2,1]

Options
A. 0.187270
B. 1.947922
C. 0.810833
D. 0.000000
Answer: D

Explanation:
Partial derivative of 'f' at (x,y) w.r.t x & y is calculated as:

dfdx =

*2*x + 2*y*(x*y - 3) - 6*

dfdy =

*2*x*(x*y - 3)*

Starting guess is taken as [2,1]
*For each iteration a new (x,y) is calculated using the step size And the formula [x(i+1),y(i+1)]=[x(i),y(i)]-step_size*grad(f)@(x(i),y(i)).*
After the given number of iterations, 'optimum_value' and 'optimum_point' is calculated from the value of function at each step

optimum_value =

0.0

optimal_point =

[2.99975, 1.0001]

<Q. 6, V. 5>

You are stuck at a high altitude on a snow mountain while skewing. There is a drone, which can capture the topography of the region around you and converts it into a mathematical expression:

f(x,y) = (x - 6.00)^2 + (4.00 - xy)^2.

You have to reach the possible lowest altitude by following a steepest decent path at every step with the step size of 0.03 and find the lowest possible altitude w.r.t mean sea level that can be reached in 200 steps.
Starting guess can be taken as [2,1]

Options
A. 0.678649
B. 0.953264
C. 0.339021
D. 0.001000
Answer: D

Explanation:
Partial derivative of 'f' at (x,y) w.r.t x & y is calculated as:

dfdx =

*2*x + 2*y*(x*y - 4) - 12*

dfdy =

*2*x*(x*y - 4)*

Starting guess is taken as [2,1]
*For each iteration a new (x,y) is calculated using the step size And the formula [x(i+1),y(i+1)]=[x(i),y(i)]-step_size*grad(f)@(x(i),y(i)).*
After the given number of iterations, 'optimum_value' and 'optimum_point' is calculated from the value of function at each step

optimum_value =

0.001

```
optimal_point =

[5.9729, 0.67255]
```

Published with MATLAB® R2021a