```
clc;
clear all;
fprintf("CH5019 - Project \nGroup no. - 25");
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

function variant(x,y)

 fprintf("\n-------------------------------------------------------------------------
\n\n");
    if y == 1
        fprintf("Q %d. - Type: Numerical\n",x);
    end
    fprintf("<Q. %d, V. %d>\n\n",x,y);
end

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

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

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

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\tA\t\t\t\t    W\t\t\t\tS\t\t\t\tV_dash\n\n");
 fprintf("\n\t A  =  W * S * V_dash\nAs: \n");
 W
 S
 V_dash
    for i = 1:n
        var=0;
        if i == 2
            var=1;
        end
        print(A,n,i,var);
        print(W,n,i,0);
        print(S,n,i,0);
        print(V_dash,n,i,0);
        fprintf("\n");
    end
    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");

    % Function to print matrices inline
    function print(X,n,i,var)
        for j = 1:n
            if X(i,j) < 0 || X(i,j) >= 10
                fprintf('');
            else
                fprintf(' ');
            end
            fprintf("%.3f ",X(i,j));
        end
        if i == 2 && var == 1
            fprintf("  =  ");
        else
            fprintf("\t ");
        end
    end
```

```
end

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

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*
*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 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. 0.357*
*B. 0.102*
*C. 0.157*
*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. 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 8.*

*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. 0.101*
*B. 0.156*
*C. 0.001*
*D. 0.356*

*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*
*= 0.001*

---

*<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 |
|---|---|
| 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. 0.157*

*C. 0.102*
*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. 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 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. 0.102*
*B. 0.357*
*C. 0.002*
*D. 0.157*

*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. 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 |
|---|---|
| 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 10.*
*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: [357 597 145 303 560 443 161 402 506 388 749 795 844 909 975 ]*

*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 , 0.0470 ]*
*[ 2 , 0.0220 ]*
*[ 3 , 0.4270 ]*
*[ 4 , 0.0050 ]*
*[ 5 , 0.1680 ]*
*[ 6 , 0.4950 ]*
*[ 7 , 0.4090 ]*
*[ 8 , 0.0620 ]*
*[ 9 , 0.4920 ]*
*[ 10 , 0.2980 ]*
*[ 11 , 0.3640 ]*
*[ 12 , 0.3500 ]*
*[ 13 , 0.0640 ]*
*[ 14 , 0.2760 ]*
*[ 15 , 0.1710 ]*

*Options*
*A. 0.7010*
*B. 0.0690*
*C. 0.5960*
*D. 0.8370*

*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 - 0.298000)*(1 - 0.364000)*(1 - 0.350000)*(1 - 0.064000)*(1 - 0.276000)*(1 - 0.171000)*
*= 0.8370*

*----------------------------------------------------------------------------*

*<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: [261 60 96 522 256 423 765 809 21 745 689 848 866 883 984 ]*

*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 , 0.4670 ]
[  2 , 0.3710 ]
[  3 , 0.0360 ]
[  4 , 0.1360 ]
[  5 , 0.0400 ]
[  6 , 0.0900 ]
[  7 , 0.2450 ]
[  8 , 0.2860 ]
[  9 , 0.3300 ]
[ 10 , 0.4710 ]
[ 11 , 0.4310 ]
```

*[ 12 , 0.3970 ]*
*[ 13 , 0.2380 ]*
*[ 14 , 0.4240 ]*
*[ 15 , 0.3690 ]*

*Options*
*A. 0.7160*
*B. 0.8700*
*C. 0.2510*
*D. 0.9050*

*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 - 0.431000)*(1 - 0.397000)*(1 - 0.238000)*(1 - 0.424000)*(1 - 0.369000)*
*= 0.9050*

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

*<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: [377 633 530 420 540 649 229 698 238 703 760 844 929 930 987 ]*

*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 , 0.3210 ]*

```
[  2 , 0.2150 ]
[  3 , 0.1970 ]
[  4 , 0.4110 ]
[  5 , 0.2330 ]
[  6 , 0.2140 ]
[  7 , 0.4010 ]
[  8 , 0.1600 ]
[  9 , 0.1880 ]
[ 10 , 0.1410 ]
[ 11 , 0.4850 ]
[ 12 , 0.4690 ]
[ 13 , 0.0060 ]
[ 14 , 0.2430 ]
[ 15 , 0.4860 ]
```

*Options*
*A.  0.7840*
*B.  0.7850*
*C.  0.1010*
*D.  0.9262*

*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 9).*

*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 - 0.188000)*(1 - 0.141000)*(1 - 0.485000)*(1 - 0.469000)*(1 - 0.006000)*(1 - 0.243000)*(1 - 0.486000)*
*= 0.9262*

---

*<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: [339 449 346 328 51 215 372 265 441 160 467 678 890 895 911 ]*

*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 , 0.1160 ]*
*[ 2 , 0.0540 ]*
*[ 3 , 0.1870 ]*
*[ 4 , 0.3790 ]*
*[ 5 , 0.1380 ]*
*[ 6 , 0.3090 ]*
*[ 7 , 0.1720 ]*
*[ 8 , 0.2760 ]*
*[ 9 , 0.4140 ]*
*[ 10 , 0.4560 ]*
*[ 11 , 0.4540 ]*
*[ 12 , 0.1190 ]*
*[ 13 , 0.0010 ]*
*[ 14 , 0.0960 ]*
*[ 15 , 0.1590 ]*

*Options*
*A. 0.5270*
*B. 0.7140*
*C. 0.5340*
*D. 0.8013*

*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 - 0.456000)*(1 - 0.454000)*(1 - 0.119000)*(1 - 0.001000)*(1 - 0.096000)*(1 - 0.159000)*
*= 0.8013*

---

*<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:   [694 500 48 74 631 532 300 624 76 628 591 733 760 846 883 ]*

*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 , 0.0800 ]
[  2 , 0.4410 ]
[  3 , 0.3590 ]
[  4 , 0.1950 ]
[  5 , 0.0470 ]
[  6 , 0.3720 ]
[  7 , 0.3390 ]
[  8 , 0.0070 ]
[  9 , 0.0800 ]
[ 10 , 0.1910 ]
[ 11 , 0.4040 ]
[ 12 , 0.1970 ]
[ 13 , 0.2270 ]
[ 14 , 0.1380 ]
[ 15 , 0.1060 ]
```

*Options*
*A.  0.1540*
*B.  0.0750*
*C.  0.6630*
*D.  0.7149*

*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 - 0.404000)*(1 - 0.197000)*(1 - 0.227000)*(1 - 0.138000)*(1 - 0.106000)*
*= 0.7149*

---

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

*A housefly in a room, travels to the point (6, 7, 8), and disorients its path by an angle of 105 degrees with respect to the positive X-axis, and finally reaches the point P(-9, 10, 8) 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.*
```
    1.7493    1.8297    3.6915
    0.7457   -2.9838   -0.5734
   -0.8815   -0.2291   -2.9204
```

*B.*
```
    7.4971    3.0137    4.6115
   22.2854   19.8156   24.7982
  -21.8854  -15.9621  -20.7966
```

*C.*
```
   11.3593    7.5107    4.4684
  -13.1963   -8.1273   -5.8971
    7.8977    4.1354    2.9724
```

*D.*
```
   15.0963   21.0171    8.4371
   -3.6622  -25.9952   -8.0087
   -1.8292   10.2982    0.3238
```

*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   -9.5391
        0    0.9659   -0.2588    4.6909
        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 =*

```
  15.0963
  -3.6622
  -1.8292
   1.0000
```

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

*result =*

```
  -3.0000  -19.8547   12.2796
```

*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 (2, 10, -4), and disorients its path by an angle of 90 degrees with respect to the positive X-*

*axis, and finally reaches the point P(2, 0, 0) 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.*

| | | |
|---|---|---|
| 1.7066 | 4.1786 | 2.9352 |
| -8.4903 | -12.9193 | -8.2030 |
| 10.5283 | 15.2024 | 9.9247 |

*B.*

| | | |
|---|---|---|
| 1.6456 | 1.7365 | 6.0524 |
| -4.0207 | -4.5790 | -5.1067 |
| -0.3136 | -0.1362 | -3.5290 |

*C.*

| | | |
|---|---|---|
| 5.8481 | 4.8650 | 7.3445 |
| -5.8005 | -4.2157 | -7.1099 |
| 2.0912 | -0.4947 | 1.3810 |

*D.*

| | | |
|---|---|---|
| 21.6908 | 5.6437 | 14.7088 |
| 2.0236 | 4.6456 | 1.6825 |
| -14.8374 | -4.7334 | -8.3363 |

*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    2.0000
         0    0.0000   -1.0000    4.0000
         0    1.0000    0.0000   10.0000
         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 =*

```
   21.6908
    2.0236
  -14.8374
    1.0000
```

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

*result =*

```
    4.0000    4.0000   10.0000
```

*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, 4, -4), and disorients its path by an angle of 75 degrees with respect to the positive X-axis, and finally reaches the point P(-2, -7, 10) 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.*

```
    9.7196   10.1941   10.2194
   -1.7623    1.0548   -3.8432
   -2.1853   -4.5598    0.2876
```

*B.*

```
    9.9253    7.2360    8.8281
   11.7301    6.0839    9.8882
  -18.7205  -12.4141  -17.3351
```

C.

```
    1.3256    0.7374    2.1378
   -4.8849   -2.5122   -4.0376
    2.9622   -0.1281    0.4749
```

D.

```
    4.0998    5.4457   25.1362
   -7.2541  -11.5033  -15.1358
   -0.3401    0.5453   -8.5134
```

*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    4.8990
         0    0.9659    0.2588    2.8284
         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 =*

```
    4.0998
   -7.2541
   -0.3401
    1.0000
```

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

*result =*

```
   -8.0000   -6.5720   -1.3449
```

*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 (0, 8, -9), and disorients its path by an angle of 105 degrees with respect to the positive X-axis, and finally reaches the point P(-8, -4, 0) 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.*

```
 -156.3088  -89.2309  -26.4446
  322.7466  183.9004   53.6603
 -148.5247  -83.3821  -21.9614
```

*B.*

```
    5.3757    1.5290    1.9475
   34.9343  107.6331  138.9779
  -28.5511  -83.8242 -108.2600
```

*C.*

```
    2.9669    2.2308    1.4515
   -2.3801   -0.6798   -1.8817
    4.4390    2.7106    4.3881
```

*D.*

```
    6.2627    2.1650    0.7312
   -3.7372  -11.1650  -11.7947
    1.7818    4.4486   10.7535
```

*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         0
         0   -0.2588   -0.9659    6.6228
         0    0.9659   -0.2588   10.0568
         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 =

    6.2627
   -3.7372
    1.7818
    1.0000
```

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

```
result =

   -8.0000    7.6581    6.1931
```

*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 (9, -1, -1), and disorients its path by an angle of 105 degrees with respect to the positive X-axis, and finally reaches the point P(-1, -8, -1) 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.*

| | | |
|---|---|---|
| 26.9868 | -55.8862 | 104.1881 |
| 0.6288 | 7.3185 | 2.5191 |
| -7.6376 | 7.8998 | -29.6974 |

*B.*

| | | |
|---|---|---|
| 3.5715 | -0.0407 | -0.3696 |
| -1.7154 | 1.7082 | -3.5012 |
| 0.7489 | 0.4654 | 5.3193 |

*C.*

| | | |
|---|---|---|
| -4.0597 | -8.9215 | 2.6458 |
| 7.0195 | 12.0971 | -0.9821 |
| -1.3052 | -1.8528 | 1.5324 |

*D.*

| | | |
|---|---|---|
| 3.6644 | -2.6966 | -11.3814 |
| -16.0634 | -6.9994 | 6.1496 |
| 4.7469 | 4.3042 | 7.4205 |

*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    9.0000
         0   -0.2588   -0.9659    1.2247
         0    0.9659   -0.2588   -0.7071
         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 =

    3.6644
  -16.0634
    4.7469
    1.0000
```

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

```
result =

    8.0000    4.2612   -8.1757
```

*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 =

    1.1351    0.2262   -1.2250    3.1961
    2.1121    3.0895    3.0074   -2.0064
    2.4826    3.5071   -0.7489   -2.2438
    1.9259   -1.6990   -0.3348   -0.6715
```

*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.*

| | | | |
|---|---|---|---|
| 0.6838 | 0.3003 | 0.2855 | 0.1958 |
| -1.3586 | 0.0618 | -0.9041 | -0.4250 |
| 0.0908 | -0.7323 | 0.1350 | -0.1388 |
| 1.4455 | 0.5310 | 1.0535 | 0.7938 |

*B.*

| | | | |
|---|---|---|---|
| 1.8721 | 5.6150 | 1.8313 | 3.1091 |
| -0.3870 | -1.0976 | -0.3719 | -0.5994 |
| -2.5580 | -11.6544 | -2.0273 | -4.6989 |
| 1.3259 | 6.6668 | 0.9405 | 2.4411 |

*C.*

| | | | |
|---|---|---|---|
| -1.1914 | -4.9288 | -1.7104 | -5.8047 |
| 0.3175 | 1.2083 | 0.3518 | 0.9395 |
| 0.7627 | 2.6260 | 1.0868 | 3.1932 |
| -0.1054 | 0.2158 | 0.0270 | 0.5877 |

*D.*

| | | | |
|---|---|---|---|
| 1.0318 | 0.5605 | 0.1753 | 0.7216 |
| 0.1853 | -0.0239 | 0.2379 | -0.9914 |
| -0.4326 | -2.2607 | 0.3341 | -2.4876 |
| -0.0961 | 2.1143 | -0.1146 | 3.0708 |

*Answer: D*
*Explanation:*

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

*A W S V_dash*

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

*W =*

| | | | |
|---|---|---|---|
| 0.2107 | 0.8731 | 0.2610 | 0.3538 |
| -0.7295 | -0.1538 | 0.2901 | 0.6001 |
| -0.6507 | 0.4559 | -0.2496 | -0.5536 |
| 0.0065 | 0.0784 | -0.8862 | 0.4565 |

*S =*

| | | | |
|---|---|---|---|
| 6.7420 | 0 | 0 | 0 |
| 0 | 3.5936 | 0 | 0 |

```
         0         0    2.7528         0
         0         0         0    2.3240


V_dash =

   -0.4308    0.5424   -0.5149    0.5051
   -0.6673    0.3306    0.5760   -0.3369
   -0.2917   -0.5287    0.3764    0.7026
    0.5329    0.5631    0.5112    0.3710

 1.135  0.226 -1.225  3.196    0.211  0.873  0.261  0.354    6.742
 0.000  0.000  0.000   -0.431  0.542 -0.515  0.505
 2.112  3.089  3.007 -2.006   =  -0.729 -0.154  0.290  0.600    0.000
 3.594  0.000  0.000   -0.667  0.331  0.576 -0.337
 2.483  3.507 -0.749 -2.244   -0.651  0.456 -0.250 -0.554    0.000
 0.000  2.753  0.000   -0.292 -0.529  0.376  0.703
 1.926 -1.699 -0.335 -0.672    0.007  0.078 -0.886  0.456    0.000
 0.000  0.000  2.324    0.533  0.563  0.511  0.371

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

   -0.4308   -0.6673   -0.2917    0.5329
    0.5424    0.3306   -0.5287    0.5631
   -0.5149    0.5760    0.3764    0.5112
    0.5051   -0.3369    0.7026    0.3710

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
   -0.9494    1.0000         0         0
    0.9312   -0.7246    1.0000         0
   -0.7943   -0.4548   -0.6962    1.0000


U =

    0.5424    0.3306   -0.5287    0.5631
         0    0.8899   -0.1254    1.0458
         0         0    1.1041    0.6044
         0         0         0    1.8766


The eigen values of the matrix U are :
eigs_of_u =
```

```
    1.8766
    1.1041
    0.8899
    0.5424
```

*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 =*

| | | | |
|---|---|---|---|
| -0.7933 | 2.2795 | -1.4272 | -3.3626 |
| -2.0582 | -0.9055 | 1.9107 | -2.1992 |
| 1.6564 | 2.6803 | 1.6379 | 4.8794 |
| -3.7567 | -3.0689 | 2.9721 | -2.0432 |

*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.*

| | | | |
|---|---|---|---|
| 0.5275 | -0.5411 | -1.0460 | -0.6921 |
| 0.2430 | 1.4985 | 1.3575 | 0.8372 |
| -0.1522 | -0.3719 | -0.2135 | -0.6904 |
| -0.0498 | -0.3295 | -0.1781 | 0.7617 |

*B.*

| | | | |
|---|---|---|---|
| 0.9982 | 0.8945 | -0.0230 | 0.0045 |
| -0.5412 | -0.4749 | 0.0332 | 0.0537 |
| 1.3099 | 1.0925 | -0.0549 | -0.0296 |
| -1.2175 | -1.0716 | 0.0289 | -0.0122 |

*C.*

| | | | |
|---|---|---|---|
| 1.4527 | 1.2086 | 1.0969 | 0.6362 |
| -0.7589 | 0.1840 | -0.4896 | -1.3490 |
| 0.1408 | 0.2138 | 0.3847 | 0.3276 |
| -0.0416 | -0.8363 | -0.1964 | 0.9485 |

*D.*

| | | | |
|---|---|---|---|
| 1.0974 | 0.2221 | 0.9948 | 0.9872 |
| 0.7165 | 1.6060 | 0.8332 | 0.6770 |
| -0.3286 | -0.4339 | -0.7063 | -1.8929 |
| -0.6476 | -0.3545 | -0.7243 | 0.6552 |

*Answer: D*

*Explanation:*

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

```
 A        W    S    V_dash


  A  =  W * S * V_dash
```

*As:*

```
W =

   -0.1941   -0.6801    0.6472    0.2845
   -0.4045    0.1666    0.4063   -0.8022
    0.6316    0.4514    0.6237    0.0911
   -0.6322    0.5532    0.1644    0.5170


S =

    8.2994         0         0         0
         0    5.3403         0         0
         0         0    2.8346         0
         0         0         0    0.4871


V_dash =

    0.5311   -0.2123   -0.3296   -0.7512
    0.4286   -0.4099    0.8024    0.0668
   -0.1615    0.6877    0.4808   -0.5195
    0.7128    0.5604   -0.1280    0.4018

-0.793  2.279 -1.427 -3.363   -0.194 -0.680  0.647  0.285    8.299
 0.000  0.000  0.000    0.531 -0.212 -0.330 -0.751
-2.058 -0.906  1.911 -2.199   =  -0.405  0.167  0.406 -0.802    0.000
 5.340  0.000  0.000    0.429 -0.410  0.802  0.067
 1.656  2.680  1.638  4.879    0.632  0.451  0.624  0.091    0.000
 0.000  2.835  0.000   -0.162  0.688  0.481 -0.520
-3.757 -3.069  2.972 -2.043   -0.632  0.553  0.164  0.517    0.000
 0.000  0.000  0.487    0.713  0.560 -0.128  0.402
```

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

```
V =

    0.5311    0.4286   -0.1615    0.7128
   -0.2123   -0.4099    0.6877    0.5604
   -0.3296    0.8024    0.4808   -0.1280
   -0.7512    0.0668   -0.5195    0.4018
```

*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
    0.4388    1.0000         0         0
    0.2826   -0.5546    1.0000         0
   -0.7071    0.6155   -0.7862    1.0000


U =

   -0.7512    0.0668   -0.5195    0.4018
         0    0.7731    0.7087   -0.3042
         0         0    1.2275    0.2781
         0         0         0    1.4029


The eigen values of the matrix U are :
eigs_of_u =

    1.4029
    1.2275
    0.7731
   -0.7512

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

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

<Q. 4, V. 3>

You are given a matrix
A =

    2.4678    4.1573    4.6772    1.5824
   -2.2261    5.4986    2.8971    2.4770
    4.3626   -0.8034   -1.3977   -0.1317
    5.7154   -1.8169   -0.7401   -3.3242

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.
    0.1246   -0.2637    0.3828   -0.0365
```

```
   -0.4329    0.4560   -0.1432   -0.2690
    1.1667    1.1244    1.3664    0.9235
   -0.2939   -0.7813   -0.8478   -0.0280
```

B.

```
    0.1507   -0.6309   -0.5725   -0.2857
   -0.1253    0.0438   -0.0710   -0.2400
    0.1130    0.4126    0.5399    0.2518
    0.1424    0.3130    0.2803    0.5296
```

C.

```
   -0.1071   -0.7718   -0.6310   -0.6155
    0.1273    0.5040    0.7491    2.0642
   -0.4893    0.0873   -0.8344   -3.9502
    0.7986    0.5423    1.0598    2.7723
```

D.

```
   -0.6383    1.9084   -0.5729    1.7027
    0.2513   -5.0097    1.3192   -3.8081
   -0.9469   -0.8292   -1.2435   -0.5225
    0.9546    6.1823   -1.0648    4.4930
```

*Answer: D*
*Explanation:*

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

```
 A        W    S    V_dash


  A  =  W * S * V_dash
```

*As:*

W =

```
    0.4095   -0.7343    0.1956   -0.5048
    0.6655   -0.1637   -0.2032    0.6993
   -0.3176   -0.3812   -0.8674   -0.0391
   -0.5372   -0.5373    0.4101    0.5046
```

S =

```
   10.3292         0         0         0
         0    7.3678         0         0
         0         0    1.9965         0
         0         0         0    1.3127
```

V_dash =

```
   -0.4770   -0.8390   -0.2531   -0.0677
    0.6383   -0.3625   -0.1766    0.6558
```

```
    0.4535   -0.4042    0.6185   -0.4984
    0.3993    0.0365   -0.7227   -0.5631

 2.468  4.157  4.677  1.582    0.409 -0.734  0.196 -0.505   10.329
 0.000  0.000  0.000   -0.477 -0.839 -0.253 -0.068
-2.226  5.499  2.897  2.477   =   0.666 -0.164 -0.203  0.699    0.000
 7.368  0.000  0.000    0.638 -0.362 -0.177  0.656
 4.363 -0.803 -1.398 -0.132   -0.318 -0.381 -0.867 -0.039    0.000
 0.000  1.997  0.000    0.454 -0.404  0.618 -0.498
 5.715 -1.817 -0.740 -3.324   -0.537 -0.537  0.410  0.505    0.000
 0.000  0.000  1.313    0.399  0.036 -0.723 -0.563
```

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

```
V =

   -0.4770    0.6383    0.4535    0.3993
   -0.8390   -0.3625   -0.4042    0.0365
   -0.2531   -0.1766    0.6185   -0.7227
   -0.0677    0.6558   -0.4984   -0.5631
```

*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
    0.5685    1.0000         0         0
    0.0807    0.8113    1.0000         0
    0.3017   -0.0796   -0.7792    1.0000


U =

   -0.8390   -0.3625   -0.4042    0.0365
         0    0.8444    0.6834    0.3785
         0         0   -1.0201   -0.8731
         0         0         0   -1.3838
```

*The eigen values of the matrix U are :*

```
eigs_of_u =

   -1.3838
   -1.0201
    0.8444
   -0.8390
```

*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 =*

| | | | |
|---|---|---|---|
| 5.7936 | 0.1749 | -3.9113 | -3.0166 |
| -0.9199 | 1.0883 | -3.9239 | -0.2809 |
| -0.9241 | 2.2123 | -0.0601 | -2.2868 |
| 1.4994 | 5.5524 | -1.4919 | 1.2939 |

*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.*

| | | | |
|---|---|---|---|
| 2.0971 | 0.6983 | 1.6754 | 2.2037 |
| 0.9667 | 1.0264 | 1.1572 | 1.4761 |
| -0.7884 | -0.2457 | -0.2453 | -1.1741 |
| -1.2620 | -0.5915 | -1.4937 | -1.3289 |

*B.*

| | | | |
|---|---|---|---|
| -0.2626 | -0.3627 | -0.2570 | -0.1252 |
| 0.9908 | 1.3177 | 0.8678 | 0.4430 |
| -0.3436 | -0.4442 | -0.2682 | -0.1489 |
| -0.4018 | -0.5505 | -0.3529 | -0.1601 |

*C.*

| | | | |
|---|---|---|---|
| 2.4662 | 1.4079 | 0.9999 | 1.0965 |
| 1.8174 | 1.2712 | 0.7542 | 0.8832 |
| -1.9305 | -1.2173 | -0.8391 | -0.9018 |
| -3.6386 | -2.2301 | -1.4817 | -1.6180 |

*D.*

| | | | |
|---|---|---|---|
| -0.3745 | -0.6031 | -1.0119 | 1.1334 |
| -0.7138 | -1.1315 | -0.5354 | -0.6955 |
| 0.7104 | 0.8227 | 1.7907 | 0.0922 |
| 0.5932 | 0.1549 | 0.3671 | -0.4031 |

*Answer: D*
*Explanation:*

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

*A W S V_dash*

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

W =

   -0.8456    0.5133    0.1299    0.0682
   -0.2827   -0.2011   -0.8894   -0.2976
   -0.1144   -0.2573   -0.2181    0.9344
   -0.4382   -0.7937    0.3801   -0.1834


S =

    8.2456         0         0         0
         0    5.8870         0         0
         0         0    3.5450         0
         0         0         0    2.8716


V_dash =

   -0.6295    0.3748    0.6607   -0.1636
   -0.3810   -0.8672    0.1925    0.2566
    0.6157   -0.0032    0.6849    0.3895
    0.2819   -0.3279    0.2394   -0.8693

 5.794  0.175 -3.911 -3.017   -0.846  0.513  0.130  0.068    8.246
 0.000  0.000  0.000   -0.629  0.375  0.661 -0.164
-0.920  1.088 -3.924 -0.281   =  -0.283 -0.201 -0.889 -0.298    0.000
 5.887  0.000  0.000   -0.381 -0.867  0.192  0.257
-0.924  2.212 -0.060 -2.287   -0.114 -0.257 -0.218  0.934    0.000
 0.000  3.545  0.000    0.616 -0.003  0.685  0.390
 1.499  5.552 -1.492  1.294   -0.438 -0.794  0.380 -0.183    0.000
 0.000  0.000  2.872    0.282 -0.328  0.239 -0.869
```

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

```
V =

   -0.6295   -0.3810    0.6157    0.2819
    0.3748   -0.8672   -0.0032   -0.3279
    0.6607    0.1925    0.6849    0.2394
   -0.1636    0.2566    0.3895   -0.8693
```

*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
    0.5673    1.0000         0         0
```

```
   -0.9527    0.2024    1.0000         0
   -0.2476   -0.3116    0.3243    1.0000


U =

    0.6607    0.1925    0.6849    0.2394
         0   -0.9764   -0.3918   -0.4637
         0         0    1.3476    0.6038
         0         0         0   -1.1504


The eigen values of the matrix U are :
eigs_of_u =

    1.3476
   -1.1504
   -0.9764
    0.6607

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 =

    0.7920   -3.9391   -3.4838    3.5794
   -2.7028    5.6673    0.7319    3.9373
   -0.1364   -3.9274    4.2842   -1.7614
   -3.2678   -3.9030    3.9849    0.8449

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.
    0.5867   -0.3152   -0.4300   -0.3438
   -0.8062   -0.4166   -1.6720   -1.4054
   -0.9230   -0.1200    0.0295   -0.5496
    1.8545    1.1356    2.3283    2.6284

B.
    0.2806   -0.5654   -1.2375   -0.7102
   -0.2819   -0.1693   -0.9260   -0.9420
    0.3654    0.8115    1.8798    0.9927
```

```
    0.0975    0.1857    0.4797    0.8224
```

C.

```
    1.0576   -0.1424    0.0701   -0.0596
   -0.8721   -0.3071   -0.2172   -1.4186
   -0.4994   -0.8461    0.7237   -1.0845
    0.6074    1.0084    0.0986    2.0857
```

D.

```
    3.6622    1.8864    4.1372    0.6275
   -3.0301   -2.0323   -2.8602   -0.3635
   -0.6532   -0.2080   -1.5596   -0.0092
   -2.9898   -1.5040   -2.3612   -1.6209
```

Answer: D
Explanation:

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

```
 A        W    S    V_dash


  A  =  W * S * V_dash
```

As:

W =

```
    0.1134   -0.6807   -0.6806   -0.2460
   -0.6134    0.5136   -0.4917   -0.3437
    0.5972    0.2903    0.0780   -0.7437
    0.5042    0.4342   -0.5375    0.5180
```

S =

```
    9.4154         0         0         0
         0    7.2190         0         0
         0         0    5.8269         0
         0         0         0    1.1154
```

V_dash =

```
    0.0020   -0.4690    0.4352   -0.7685
   -0.8748    0.3820    0.2893   -0.0716
    0.3955    0.7925    0.0349   -0.4629
   -0.2799   -0.0774   -0.8519   -0.4359
```

```
 0.792 -3.939 -3.484  3.579    0.113 -0.681 -0.681 -0.246    9.415
 0.000  0.000  0.000    0.002 -0.469  0.435 -0.769
-2.703  5.667  0.732  3.937   =  -0.613  0.514 -0.492 -0.344    0.000
 7.219  0.000  0.000   -0.875  0.382  0.289 -0.072
```

```
-0.136 -3.927  4.284 -1.761    0.597  0.290  0.078 -0.744    0.000
 0.000  5.827  0.000    0.395  0.793  0.035 -0.463
-3.268 -3.903  3.985  0.845    0.504  0.434 -0.538  0.518    0.000
 0.000  0.000  1.115   -0.280 -0.077 -0.852 -0.436
```

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

*V =*

```
    0.0020   -0.8748    0.3955   -0.2799
   -0.4690    0.3820    0.7925   -0.0774
    0.4352    0.2893    0.0349   -0.8519
   -0.7685   -0.0716   -0.4629   -0.4359
```

*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
   -0.0026    1.0000         0         0
    0.6103   -0.4865    1.0000         0
   -0.5663   -0.2843   -0.0909    1.0000
```

*U =*

```
   -0.7685   -0.0716   -0.4629   -0.4359
         0   -0.8750    0.3943   -0.2810
         0         0    1.2668    0.0519
         0         0         0   -1.1739
```

*The eigen values of the matrix U are :*

*eigs_of_u =*

```
    1.2668
   -1.1739
   -0.8750
   -0.7685
```

*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 = 0.37*(x(2) - x(1)^2)^2 + (0.71 - x(1))^2 Joules.*

```
It is defined in the region such that:
   x(1) + 0.42*x(2) <= 0.30
   0.68*x(1) + x(2) = 0.34
   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.  1.9481 0.9219
B.  0.1147 0.0138
C.  0.9133 0.0336
D.  0.2198 0.1862

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.37*(q - x(1)^2)^2 + (0.71 - 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    0.4245


b =

    0.2988


Aeq =
```

```
    0.6776    1.0000


beq =

    0.3351


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.2198    0.1862


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

<Q. 5, V. 2>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.53*(x(2) - x(1)^2)^2 + (0.74 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.69*x(2) <= 0.55
   0.33*x(1) + x(2) = 0.58
   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.  1.8974 0.2644
B.  0.8902 0.6676
C.  0.3038 0.3079
D.  0.1871 0.5181

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.53*(q - x(1)^2)^2 + (0.74 - 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    0.6935


b =

    0.5463


Aeq =

    0.3251    1.0000


beq =

    0.5789


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.1871    0.5181


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

<Q. 5, V. 3>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.63*(x(2) - x(1)^2)^2 + (0.56 - x(1))^2 Joules.

It is defined in the region such that:
```

```
   x(1) + 0.10*x(2) <= 0.38
   0.57*x(1) + x(2) = 0.05
   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.  1.4800 0.2423
B.  0.0233 0.8114
C.  0.4398 0.4393
D.  0.3468 -0.1467

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.63*(q - x(1)^2)^2 + (0.56 - 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    0.0997


b =

    0.3814


Aeq =

    0.5676    1.0000
```

```
beq =

    0.0502


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.3468   -0.1467


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

<Q. 5, V. 4>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.89*(x(2) - x(1)^2)^2 + (0.10 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.94*x(2) <= 0.37
   0.16*x(1) + x(2) = 0.01
   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.  1.3663 0.0171
B.  0.4299 0.3075
C.  0.4266 0.2703
D.  0.1006 -0.0015

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.89*(q - x(1)^2)^2 + (0.10 - 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    0.9369


b =

    0.3735


Aeq =

    0.1615    1.0000


beq =

    0.0148


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.1006   -0.0015


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

<Q. 5, V. 5>

The energy of a particle in the 2D coordinate system is defined as
  E = 0.87*(x(2) - x(1)^2)^2 + (0.72 - x(1))^2 Joules.

It is defined in the region such that:
   x(1) + 0.23*x(2) <= 0.31
   0.98*x(1) + x(2) = 0.94
```

```
   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.  1.7730 0.1423
B.  0.8311 0.5043
C.  0.0558 0.6743
D.  0.1257 0.8185

Answer : D

 Explanation:
 The energy of particle in 2D coordinate system is given as
  E = 0.87*(q - x(1)^2)^2 + (0.72 - 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    0.2299


b =

    0.3139


Aeq =

    0.9790    1.0000
```

```
beq =

    0.9415


lb =

     0     0

 Solving this using the function 'fmincon' gives

x =

    0.1257    0.8185


-----------------------------------------------------------------------------------
```

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 - 2.00)^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 0.07 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. 1.072998
B. 0.177926
C. 0.416721
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 - 2) - 4


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*

*optimal_point =*

*[2, 1]*

---

*<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 - 2.00)^2 + (1.00 - 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. 1.377517*
*B. 0.991162*
*C. 0.892327*
*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 - 1) - 4*

*dfdy =*

*2*x*(x*y - 1)*

*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 =*

*[1.99972, 0.50012]*

---

*<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 - 3.00)^2 + (1.00 - xy)^2.*

*You have to reach the possible lowest altitude by following a steepest decent path at every step with the step size of 0.06 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. 0.851995*
*B. 0.012990*
*C. 1.377366*
*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 - 1) - 6*

*dfdy =*

*2*x*(x*y - 1)*

*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 =*

*[2.99978, 0.33334]*

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

*<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 - 6.00)^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 0.01 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. 1.107192*
*B. 1.414679*
*C. 1.535785*
*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) - 12*

*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 =*

*[5.98947, 0.83484]*

---

*<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 - 2.00)^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 0.05 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. 0.065367*
*B. 1.516331*
*C. 1.216976*
*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 - 4) - 4*

*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 =

0.0


optimal_point =

[2.00045, 1.9995]
```

*Published with MATLAB® R2021a*