REFERENCES

[1] A. Amaravathy, K. Thiagarajan and S. Vimala, "MDMA Method- An Optimal Solution for Transportation Problem", Middle-East Journal of Scientific Research, 2016, 24 (12): 3706-3710.

[2] A. R. Khan, A. Vilcu, N. Sultana, and S. S. Ahmed, "Determination of initial basic feasible solution of a transportation problem: a TOCM-SUM approach," no. Lxv, 2015.

[3] Charnes and Cooper, The Stepping-Stone method for explaining linear programming, Calculation in transportation problems, Management Science, 1(1)(1954), 49-69

[4] Dantzig, Application of the simplex method to a transportation problem, Activity Analysis of production and allocation, (T.C. Koopmans ed.), Wiley, New York, (1951), 359-373

[5] Dantzig, G.B. Linear Programming and Extensions, Princeton, NJ: Princeton University Press,1963.

[6] G. Monge. Mémoiresur la théorie des déblais et des remblais. Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année, 1781, pages 666–704.

[7] Goyal, S. K, A joint economic-lot-size model for purchaser and vendor: A Comment. Decision Sciences. 1988, 19, 236-41.

[8] Goyal, S.K, Improving VAM for the Unbalanced Transportation Problem. Journal Opl,1984, Res Soc 35,1113-14.

[9] H.A. Taha, Operation Research: An Introduction, 8th ed., Prentice-Hall of India, 2006.

[10] Hitchcock, F.L., The distribution of a product from several sources to numerous locations. Journal of Mathematical Physics. 1941, 20, 224–30.

[11] Hlayel and M. A. Alia, "Solving Transportation Problems Using the Best Candidates Method," *Comput.Sci. Eng. An Int. J.*, 2012, vol. 2, no. 5, pp. 23–30.

[12] Jain, C. Author, and S. Sood, "The maximum difference method to find initial basic feasible solution for transportation problem", 2015, vol.3, no. 7. pp.8–11.

[13] Justice Kwame Appati, Gideon Kwadwo Gogovi, Gabriel Obed Fosu, MATLAB Implementation of Vogel's Approximation and the Modified Distribution Methods, An international journal of advanced computer technology, 4 (1), 2015, Volume-IV, Issue-I.

[14] Kantorovich, L.V, "Mathematical Methods of Organizing and Planning Production". Management Science. 1939, 6 (4): pp. 366–422.

[15] Monalisha Pattnaik, "Transportation Problem By Monalisha's Approximation Method For Optimal Solution (MAMOS)", Log Forum, 2015, vol. 11, no. 3, pp. 267–273.

[16] Mollah Mesbahuddin Ahmed, Abu Sadat Muhammed Tanvir , Shirin SultanaSultan Mahmud and Md. Sharif Uddin, "An Effective Modification to Solve Transportation Problems:A Cost Minimization Approach", Annals of Pure and Applied Mathematics, 2014, Vol. 6, No. 2, pp. 199-206 .

[17] Mollah Mesbahuddin Ahmed, Aminur Rahman Khan, Md. Sharif Uddin, Faruque Ahmed, "A New Approach to Solve Transportation Problems", Open Journal of Optimization, 2015, 5, pp. 22-30.

[18] M.Wali Ullah, M.Alhaz Uddin and Rijwana Kawser, A Modified Vogel's Approximation Method for Obtaining a Good Primal Solution of Transportation Problems", Annals of Pure and Applied Mathematics, 2016, Vol. 11, No. 1, pp.63-71.

[19] Paul S. Dwyeri, The Direct Solution of the Transportation Problem with Reduced Matrices, MANAGEMENT SCIENCE, 1966, Vol. 13, No. 1.

[20] Purnima Adhikari , Gyan Bahadur Thapa, A Note on Feasibility and Optimality of Transportation Problem, Journal of the Institute of Engineering, Vol. 10, No. 1, pp. 59–68.

[21] Ramakrishnan. Discussed some improvement to Goyal's Modified Vogel's Approximation method for Unbalanced Transportation Problem. J Opl Res Soc.1988,39, pp. 609-10.

[22] Reinfeld, N.V. and W.R. Vogel, Mathematical Programming. Englewood Gliffs, New Jersey: Prentice-Hall,1958.

[23] Reena. G. Patel , P.H. Bhathawala,"An Innovative Approach to optimum solution of a Transportation Problem",International Journal of Innovative Research in Science, Engineering and Technology, 2016, Vol. 5, Issue 4.

[24] S.M. Abul Kalam Azad1, Md. Bellel Hossain, "A New Method for Solving Transportation Problems Considering Average Penalty", IOSR Journal of Mathematics (IOSR-JM), 2017, Volume 13, Issue 1 Ver. IV , pp 40-43.

[25] Sultan, A. & Goyal, S.K., Resolution of Degeneracy in Transportation Problems. Journal Operational Research Society. 1988, 39, pp. 411-13.

[26] Utpal Kanti Das , Md. Ashraful Babu, Aminur Rahman Khan, Md. Abu Helal, Dr. Md. Sharif Uddin, Logical Development Of Vogel's Approximation Method (LD-VAM): An Approach To Find Basic Feasible Solution Of Transportation Problem, International Journal of Scientific & Technology Research,2014, volume 3, issue 2.

APPENDIX

Appendix-01 MATLAB 2015 Code

Code for MODI Method

```
function [ x,tc ] = modi( x,a)

    b=a;
    r=size(a,1);
    c=size(a,2);

    lc=0;
    k=0;
     for i=1:r
         for j=1:c
             if(x(i,j)~=0)
                 k=k+1;
             end
         end
     end


    nbc=zeros(r,c);
    loop=zeros(r,c);

    u=zeros(r,1);
    v=zeros(1,c);
iter=0;
    mi=1;
    mj=1;

     while(k==r+c-1)
    %      FOR BASIC CELL
    %      Counting the no.of allocations in row & column */
        for i=1:r
         for j=1:c
          if x(i,j)~=0
              u(i)=u(i)+1;
          end
         end
        end

        for j=1:c
            for i=1:r
                if x(i,j)~=0
                    v(j)=v(j)+1;
                end
            end
        end
```

```
%  Selecting the row or column having max no.of allocations

    max=0;
    flag=0;
    for i=1:r
        if max<u(i)
            max=u(i);
            mi=i;
            flag=1;
        end
    end

    for j=1:c
        if max<v(j)
            max=v(j);
            mj=j;
            flag=2;
        end
    end
u=zeros(r,1);
v=zeros(1,c);

%       Assigning value for u and v
    if(flag==1)
        for j=1:c
            if x(mi,j) ~=0
                v(j)=b(mi,j);
            end
        end

        for k=1:r
            for i=1:r
                for j=1:c
                        if (x(i,j)~=0 && v(j)~=0)
                            u(i)=b(i,j)-v(j);
                        end
                end
            end

            for j=1:c
                for i=1:r
                        if(x(i,j)~=0 && u(i)~=0)
                            v(j)=b(i,j)-u(i);
                        end
                end
            end
        end
    end

if(flag==2)
        for i=1:r
            if (x(i,mj)~=0)
                u(i)=b(i,mj);
            end
        end
```

```
        for k=1:r
            for j=1:c
                for i=1:r
                    if(x(i,j)~=0 && u(i)~=0)
                        v(j)=b(i,j)-u(i);
                    end
                end
            end

            for i=1:r
                for j=1:c
                    if(x(i,j)~=0 && v(j)~=0)
                        u(i)=b(i,j)-v(j);
                    end
                end
            end
        end
end

%     FOR NON BASIC CELL
    max=0;
    for i=1:r
        for j=1:c
            if(x(i,j)==0)
                nbc(i,j)=b(i,j)-(u(i)+v(j));
                if(max>nbc(i,j))
                    max=nbc(i,j);
                    mi=i;
                    mj=j;
                end
            end
        end
    end

    if(max>=0)
      break;
    end


%        Loop Formation

    for i=1:r
        for j=1:c
            if(x(i,j)~=0)
                loop(i,j)=1;
            else
                loop(i,j)=0;
            end
            sign(i,j)=' ';

        end

    end

    for k=1:r
        for i=1:r
```

```
                for j=1:c
                    if(loop(i,j)==1)
                        lc=lc+1;
                    end
                end

                if(lc==1 && i~=mi)
                    for j=1:c
                        loop(i,j)=0;
                    end
                end
                lc=0;
            end

            lc=0;
            for j=1:c
                for i=1:r
                    if(loop(i,j)==1)
                        lc=lc+1;
                    end
                end

                if(lc==1 && j~=mj)
                    for i=1:r
                        loop(i,j)=0;
                    end
                end
                lc=0;
            end
        end



  %     Assigning the Sign
        sign(mi,mj)='+';
        i=mi;
        for k=1:15%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            for j=1:c
                if(loop(i,j)==1 && sign(i,j)==' ')
                    sign(i,j)='-';
                    break;
                end
            end

            for i=1:r
                if(loop(i,j)==1 && sign(i,j)==' ')
                    sign(i,j)='+';
                    break;
                end
            end
        end
sign;
        %  Finding @ Value
        min=9999;

        for i=1:r
```

```
            for j=1:c
                if(sign(i,j)=='-' && min>x(i,j))
                        min=x(i,j);
                end
            end
        end

        for i=1:r
             for j=1:c
                 if(sign(i,j)=='+')
                     x(i,j)=x(i,j)+min;
                 elseif(sign(i,j)=='-')
                     x(i,j)=x(i,j)-min;
                 end
             end
        end

        % Checking m+n-1 Condition
        k=0;
         for i=1:r
             for j=1:c
                 if(x(i,j)~=0)
                     k=k+1;
                 end
             end
         end
                 iter=iter+1;
                 k;
    end   % End of While

    disp('The Optimum Solution Using Modi Method');
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    x
    disp('Total Cost:');
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    tc= sum(sum(x.*a))
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    disp('Total Number iteration:');
    iter


end
```

Code for Cumulative Difference Method

```
function [ Solution, OverallCost ] = Newmtd(CostsMtx,
resources_col,demands_row)

C_startcost =CostsMtx;

Cumalative_Mat=Ccost(CostsMtx);


C_startcost =CostsMtx;

C_start = Cumalative_Mat;
C = C_start;
m = size(C,1);
n = size(C,2);
a = resources_col;
b = demands_row;
X = zeros(m,n);

stop = 0;

while stop == 0

for i = 1:m
for j = 1:n

if a(i,1) == 0
   C(i,j) = min(C(:,j));
end
if b(1,j) == 0
    C(i,j) = min(C(i,:));
end
end
end
C_sort_col = sort(C,1,'descend');
C_sort_row = sort(C,2,'descend');

Diff_customer = abs(C_sort_col(1,:) - C_sort_col(2,:));
Diff_supplier = abs(C_sort_row(:,1) - C_sort_row(:,2));

for i = 1:m
if a(i,1) == 0
Diff_supplier(i,1) = 0;
end
end
for j = 1:n
if b(1,j) == 0
Diff_customer(1,j) = 0;
end
end
Max_Diff_customer = max(Diff_customer);
Max_Diff_supplier = max(Diff_supplier);
```

```
Customer_nr =
find(Diff_customer==max(Max_Diff_customer,Max_Diff_supplier));
Supplier_nr =
find(Diff_supplier==max(Max_Diff_customer,Max_Diff_supplier));

if isempty(Customer_nr) == 0
Supplier_nr_ = find(C(:,Customer_nr(1)) ==
max(C(:,Customer_nr(1))));
X(Supplier_nr_(1),Customer_nr(1)) =
min(a(Supplier_nr_(1),1),b(1,Customer_nr(1)));

a(Supplier_nr_(1),1) = a(Supplier_nr_(1),1) -
X(Supplier_nr_(1),Customer_nr(1));
b(1,Customer_nr(1)) = b(1,Customer_nr(1)) -
X(Supplier_nr_(1),Customer_nr(1));
Supplier_nr = [];
end
if isempty(Supplier_nr) == 0
Customer_nr_ = find(C(Supplier_nr(1),:) ==
max(C(Supplier_nr(1),:)));
X(Supplier_nr(1),Customer_nr_(1)) =
min(a(Supplier_nr(1),1),b(1,Customer_nr_(1)));

a(Supplier_nr(1),1) = a(Supplier_nr(1),1) -
X(Supplier_nr(1),Customer_nr_(1));
b(1,Customer_nr_(1)) = b(1,Customer_nr_(1)) -
X(Supplier_nr(1),Customer_nr_(1));
end
a1 = a > 0;
b1 = b > 0;
if sum(a1) == 1
stop = 1;
for j = 1:n
if b(j) > 0;
X(a1 == 1,j) = b(j);
end
end
end
if sum(b1) == 1
stop = 1;
for i = 1:m
if a(i) > 0;
X(i,b1 == 1) = a(i);
end
end
end
end
Solution = X;
OverallCost = sum(sum(C_startcost .* X));

    disp('The Initial Feasible Solution Using New Method');
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    X
    disp('Initial Cost:');
    OverallCost
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
 end
```

Code Vogel's Approximation Method

```
function [ Solution, OverallCost ] = VAM(CostsMtx, resources_col,
demands_row)

[ CostsMtx,ss,sd] = inputFun(CostsMtx, resources_col, demands_row);

C_start = CostsMtx;
C = C_start;
m = size(C,1);
n = size(C,2);
a = resources_col;
b = demands_row;
X = zeros(m,n);

stop = 0;

while stop == 0

for i = 1:m
for j = 1:n
if a(i,1) == 0
C(i,j) = max(C(:,j));
end
if b(1,j) == 0
C(i,j) = max(C(i,:));
end
end
end
C_sort_col = sort(C,1);
C_sort_row = sort(C,2);

Diff_customer = abs(C_sort_col(1,:) - C_sort_col(2,:));
Diff_supplier = abs(C_sort_row(:,1) - C_sort_row(:,2));

for i = 1:m
if a(i,1) == 0
Diff_supplier(i,1) = 0;
end
end

for j = 1:n
if b(1,j) == 0
Diff_customer(1,j) = 0;
end
end

Max_Diff_customer = max(Diff_customer);
Max_Diff_supplier = max(Diff_supplier);
```

```
Customer_nr =
find(Diff_customer==max(Max_Diff_customer,Max_Diff_supplier));
Supplier_nr =
find(Diff_supplier==max(Max_Diff_customer,Max_Diff_supplier));

if isempty(Customer_nr) == 0
Supplier_nr_ = find(C(:,Customer_nr(1)) ==
min(C(:,Customer_nr(1))));
X(Supplier_nr_(1),Customer_nr(1)) =
min(a(Supplier_nr_(1),1),b(1,Customer_nr(1)));

a(Supplier_nr_(1),1) = a(Supplier_nr_(1),1) -
X(Supplier_nr_(1),Customer_nr(1));
b(1,Customer_nr(1)) = b(1,Customer_nr(1)) -
X(Supplier_nr_(1),Customer_nr(1));
Supplier_nr = [];
end
if isempty(Supplier_nr) == 0
Customer_nr_ = find(C(Supplier_nr(1),:) ==
min(C(Supplier_nr(1),:)));
X(Supplier_nr(1),Customer_nr_(1)) =
min(a(Supplier_nr(1),1),b(1,Customer_nr_(1)));

a(Supplier_nr(1),1) = a(Supplier_nr(1),1) -
X(Supplier_nr(1),Customer_nr_(1));
b(1,Customer_nr_(1)) = b(1,Customer_nr_(1)) -
X(Supplier_nr(1),Customer_nr_(1));
end

%Stop condition:

if (max(a)==0||max(b)==0)
     stop = 1;
end
a1 = a > 0;
b1 = b > 0;
if sum(a1) == 1
stop = 1;
for j = 1:n
if b(j) > 0;
X(a1 == 1,j) = b(j);
end
end
end
if sum(b1) == 1
stop = 1;
for i = 1:m
if a(i) > 0;
X(i,b1 == 1) = a(i);
end
end
end
end

 if (isempty(a1) || isempty(b1))
      stop = 1;
```

```
 end

Solution = X;
OverallCost = sum(sum(C_start .* X));
a;
b;

    disp('The Initial Feasible Solution Using Vogels Method');
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    X
    disp('Initial Cost:');
    OverallCost
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
end
```

Code for North-West Corner Rule Method

```
function [x,tc]=nwc(a,s,d)
  [ a,ss,sd] = inputFun(a,s,d);
  r=size(a,1);
  c=size(a,2);


    x=zeros(size(a));
    k=0;i=1;j=1;
    while(k<(r+c)-1)
        if(s(i)>d(j))
            k=k+1;
            x(i,j)=d(j);
            s(i)=s(i)-d(j);
            ss=ss-d(j);
            sd=sd-d(j);
            d(j)=0;
            j=j+1;
        elseif(s(i)<d(j))
            k=k+1;
            x(i,j)=s(i);
            d(j)=d(j)-s(i);
            ss=ss-s(i);
            sd=sd-s(i);
            s(i)=0;
            i=i+1;
        else
            k=k+1;
            x(i,j)=s(i);
            ss=ss-s(i);
            sd=sd-s(i);
            s(i)=0;
            d(j)=0;
            i=i+1;
            j=j+1;
        end
```

```
        if((ss==0) && (sd==0))
            break;
        end
    end
    tc= sum(sum(x.*a));

    disp('The Initial Feasible Solution Using North-West Corner
Method');
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    x
    disp('Initial Cost:');
    tc
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
end
```

Code for Least Cost Method

```
function [x,tc]=lcm(a,s,d)
  [ a,ss,sd] = inputFun(a,s,d);
  r=size(a,1);
  c=size(a,2);
  x=zeros(size(a));
  b=a;


  for i=1:r
      for j=1:c
          b(i,j)=a(i,j);
      end
  end

  k=0;mi=1;mj=1;
  while(k<(r+c)-1)
      min=9999;
      for i=1:r
          for j=1:c
              if(min>b(i,j) && b(i,j)~=-1)
                  min=b(i,j);
                  mi=i;
                  mj=j;
              end
          end
      end
    if(s(mi)>d(mj))
        k=k+1;
        x(mi,mj)=d(mj);
        s(mi)=s(mi)-d(mj);
        ss=ss-d(mj);
        sd=sd-d(mj);
        d(mj)=0;
        for i=1:r
            b(i,mj)=-1;
```

```
        end
    end

    if(s(mi)<d(mj))
        k=k+1;
        x(mi,mj)=s(mi);
        d(mj)=d(mj)-s(mi);
        ss=ss-s(mi);
        sd=sd-s(mi);
        s(mi)=0;
        for j=1:c
            b(mi,j)=-1;
        end
    end

    if(s(mi)==d(mj))
        k=k+1;
        x(mi,mj)=s(mi);
        ss=ss-s(mi);
        sd=sd-s(mi);
        s(mi)=0;
        d(mj)=0;
        for i=1:r
            b(i,mj)=-1;
        end

        for j=1:c
            b(mi,j)=-1;
        end
    end

    if((ss==0)&&(sd==0))
      break;
    end
  end
  tc= sum(sum(x.*a));

    disp('The Initial Feasible Solution Using Least Cost Method');
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
    x
    disp('Initial Cost:');
    tc
    disp('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
end
```

Appendix-02 Problem set

20 Selected problems

```
01
a=[12 4 13 18 9 2; 9 16 10 7 15 11; 4 9 10 8 9 7;  9 3 12 6 4 5;
    7 11 5 18 2 7; 16 8 4 5 1 7 ]
s=[120;80;50;90;100;60]
d=[75;85;140;40;95;65]'

02
a=[3 6 8 4;  6 1 2 5;7 8 3 9 ]
s=[20;28;17]
d=[15;19;13;18]'

03
a=[3 1 7 4; 2 6 5 9;8 3 3 2 ]
s=[300;400;500]
d=[250;350;400;200]'

04
a=[2 3 11 7; 1 0 6 1; 5 8 15 9]
s=[6;1;10]
d=[7;5;3;2]'

05
a=[4 3 5;6 5 4;8 10 7]
s=[90; 80; 100]
d=[70; 120; 80]'

06
a=[4 5 8 4; 6 2 8 1;8 7 9 10]
s=[52;57;54] d=[60;45;8;50]'

07
a=[7 5 9 11;  4 3 8 6;3 8 10 5;  2 6 7 3]
s=[30;25;20;15]
d=[30;30;20;10]'

08
a=[50 60 100 50;80 40 70 50; 90 70 30 50]
s=[20;38;16]
d=[10;18;22;24]'

09
a=[2 7 4 ; 3 3 1 ;   5 4 7]
s=[5;8;7;14]
d=[7;9;18]'

10
a=[8 6 10 9;9 12 13 7 ; 14 9 16 5]
s=[35;50;40]
d=[45;20;30;30]'
```

```
11
a=[5 2 4 1 0; 5 2 1 4 0 ;6 4 8 2 0; 4 6 5 4 0;2 8 4 5 0]
s=[30;20;12;46;46]
d=[30;50;30;27;17]'

12
a=[0 2 20 11;12 7 9 20 ;4 14 16 18]
s=[15;25;10]
d=[5;15;15;15]'

13
a=[6 3 8 7; 8 5 2 4 ; 4 9 8 4; 7 8 5 6]
s=[110;60;54;3;27]
d=[20;70;78;86]'

14
a=[4 19 22 11;1 9 14 14 ; 6 6 16 14]
s=[100;30;70]
d=[40;20;60;80]'

15
a=[2 7 4; 3 3 1 ; 5 4 7;1 6 2]
s=[5;8;7;14]
d=[7;9;18]'


16
a=[6 8 10;7 11 11;4 5 12]
s=[150;175;275]
d=[200;100;300]'

17
a=[14 19 10 7 16;6 7 7 14 12;5 14 17 9 6;13 12 9 15 8;17 5 13 16 19]
s=[30;20;12;46;46]
d=[30;50;30;27;17]'

18
a=[17 14 11 29 11;17 7 15 29 14;13 25 7 19 26;28 15 8 6 5]
s=[55;45;30;50]
d=[40;20;50;30;40]'


19
a=[31 27 20  23;30 20 19 39;20 36 20 26;22 20 26 19]
s=[6;8;16;15]
d=[9;10;12;14]'


20
a=[42 33 25 29; 41 38 45 35;26 38 28 32;33 31 27 44; 36 34 32  44]
s=[30;20;12;30;46]
d=[31;50;30;27]'
```

28 Selected problems from literature

```
1
a=[10 30 25 15;20 15 20 10;10 30 20 20;30 40 35 45]
s=[14;10;15;12]
d=[10;15;2;15]'

%2
a=[2 7 4;3 3 1;5 4 7;1 6 2]
s=[5;8;7;14]
d=[7;9;18]'

3
a=[10 2 16 14 10;6 18 12 13 16;8 4 14 12 10;14 22 20 8 18]
s=[300;500;825;375]
d=[350;400;250;150;400]'

4
a=[3 6 8 4;6 1 2 5;7 8 3 9]
s=[20;28;17]
d=[15;19;13;18]'

5
a=[6 10 14;12 19 21;15 14 17]
s=[50;50;50]
d=[30;40;55]'

6
a=[12 4 13 18 9 2;9 16 10 7 15 11;4 9 10 8 9 7;9 3 12 6 4 5;7 11 5
      18 2 7;16 8 4 5 1 10]
s=[120;80;50;90;100;60]
d=[75;85;140;40;95;65]'

7
a=[25 55 40 60;35 30 50 40;36 45 26 66;35 30 41 150]
s=[60;80;160;150]
d=[90;100;120;140]'

8
a=[9 12 9 6 9 10 ;7 3 7 7 5 5;6 5 9 11 3 11;6 8 11 2 2 10]
s=[5;6;2;9]
d=[4;4;6;2;4;2]'

9
a=[21 16 25 13;17 18 14 23; 32 27 18 41]
s=[11;13;19]
d=[6;10;12;15]'

10
a=[6 4 4 7 5;5 6 7 4 8;3 4 6 3 4]
s=[100;125;175]
d=[60;80;85;105;70]'
```

```
11
a=[2 3 11 7;1 0 6 1;5 8 15 9]
s=[6;1;10]
d=[7;5;3;2]'

12
a=[19 30 50 10;70 30 40 60;40 8 70 20]
s=[7;9;18]
d=[5;8;7;14]'

13
a=[4 19 22 11;1 9 14 14;6 6 16 14]
s=[100;30;70]
d=[40;20;60;80]'

14
a=[4 5 8 4;6 2 8 1;8 7 9 10]
s=[52;57;54]
d=[60;45;8;50]'

15
a=[6 3 8 7;8 5 2 4; 4 9 8 4; 7 8 5 6]
s=[110;60;54;30]
d=[20;70;78;86]'

16
a=[5 2 4 1;5 2 1 4; 6 4 8 2; 4 6 5 4;2 8 4 5]
s=[30;20;12;30;46]
d=[31;50;30;27]'

16
a=[3 15 17;45 30 30;13 25 42]
s=[580;240;330]
d=[310;540;300]'

17
a=[3 1 7 4;2 6 5 9;8 3 3 2]
s=[300;400;500]
d=[250;350;400;200]'

18
a=[50 60 100 50;80 40 70 50;90 70 30 50 ]
s=[20;38;16]
d=[10;18;22;24]'

19
a=[7 5 9 11;4 3 8 6;3 8 10 5;2 6 7 3]
s=[30;25;20;15]
d=[30;30;20;10]'

20
a=[4 3 5;6 5 4;8 10 7]
s=[90;80;100]
d=[70;120;80]'
```

```
21
a=[11 9 6;12 14 12;10 8 10]
s=[40;50;40]
d=[55;45;30]'

22
a=[13 18 30 8;55 20 25 40;30 6 50 10]
s=[8;10;11]
d=[4;7;6;12]'

23
a=[3 4 6 8 9;2 10 1 5 8;7 11 20 40 3;2 1 9 14 16]
s=[20;30;15;13]
d=[40;6;8;18;6]'

24
a=[12 4 9 5 9;8 1 6 6 7;12 4 7 7;10 15 6 9 1]
s=[55;45;30;50]
d=[40;20;50;30;40]'

25
a=[12 4 9 5 9;8 1 6 6 7;1 12 4 7 7; 10 15 6 9 1]
s=[55;45;30;50]
d=[40;20;50;30;40]'

6
a=[9 8 5 7;4 6 8 7;5 8 9 5]
s=[12;14;16]
d=[8;18;13;3]'

27
a=[2 2 2 1;10 8 5 4;7 6 6 8]
s=[3;7;5]
d=[4;3;4;4]'

28
a=[4 3 5;6 5 4;8 10 7]
s=[9;8;10]
d=[7;12;8]'
```