

```

#include<bits/stdc++.h>
using namespace std;
int mat[20][20];
int temp[20][20];
int base[20][20];
int n;

int check(int visited[],int size,int c){
    for(int i=0; i<size; i++){
        if(visited[i]==c){
            return 0;
        }
    }
    return 1;
}

void make_inf(int arr[],int size,int col){
    for(int i=0;i<size; i++){
        int r=arr[i];
        for(int j=0;j<n;j++){
            temp[r][j]=999;
        }
    }
    for(int j=0;j<n;j++){
        temp[j][col]=999;
    }
    int first=arr[0];
    temp[col][first]=999;
    for(int i=1; i<size; i++){
        int t=arr[i];
        temp[t][first]=999;
    }
}

int minimization(int cost){
    //Row Reduction
    int row_red=0;
    for(int i=0; i<n; i++){
        int min1=temp[i][0];
        for(int j=1; j<n; j++){
            min1=min(min1,temp[i][j]);
        }
        if(min1 != 999){
            row_red+=min1;
            for(int j=0; j<n; j++){
                if(temp[i][j] != 999){
                    temp[i][j]-=min1;
                }
            }
        }
    }
    //Column Reduction
    int col_red=0;
    for(int i=0; i<n; i++){
        int min2=temp[0][i];
        for(int j=1; j<n; j++){
            min2=min(min2,temp[j][i]);
        }
        if(min2 != 999){
            col_red+=min2;
        }
    }
}

```

```

        for(int j=0; j<n; j++){
            if(temp[j][i] != 999){
                temp[j][i]-=min2;
            }
        }
    }
    return cost+row_red+col_red;
}

int main(){
    cout<<"Enter no. of vertices"<<endl;
    cin>>n;
    cout<<"\nEnter matrix"<<endl;
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            cin>>mat[i][j];
        }
    }
    //Row Reduction
    int row_red=0;
    for(int i=0; i<n; i++){
        int min1=mat[i][0];
        for(int j=1; j<n; j++){
            min1=min(min1,mat[i][j]);
        }
        row_red+=min1;
        for(int j=0; j<n; j++){
            if(mat[i][j] != 999){
                mat[i][j]-=min1;
            }
        }
    }
    //Column Reduction
    int col_red=0;
    for(int i=0; i<n; i++){
        int min2=mat[0][i];
        for(int j=1; j<n; j++){
            min2=min(min2,mat[j][i]);
        }
        col_red+=min2;
        for(int j=0; j<n; j++){
            if(mat[j][i] != 999){
                mat[j][i]-=min2;
            }
        }
    }
    cout<<"\nafter Reduction"<<endl;
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            base[i][j]=mat[i][j];
        }
    }
    int tot=row_red+col_red;
    int visited[n];
    int size=0;
    visited[size]=0;
    size++;
    while(true){
        int mincost=INT_MAX;
        int min_ind;

```

```

for(int j=1; j<n; j++){
    int ch=check(visited,size,j);
    if(ch==1){
        for(int a=0; a<n; a++){
            for(int b=0; b<n; b++){
                temp[a][b]=base[a][b];
            }
        }
        make_inf(visited,size,j);
        int cost=minimization(tot);
        cost= cost + base[visited[size-1]][j];
        if(mincost > cost){
            mincost=cost;
            min_ind=j;
        }
    }
}
for(int a=0; a<n; a++){
    for(int b=0; b<n; b++){
        temp[a][b]=base[a][b];
    }
}
make_inf(visited,size,min_ind);
int tpp=minimization(tot);
for(int a=0; a<n; a++){
    for(int b=0; b<n; b++){
        base[a][b]=temp[a][b];
    }
}
visited[size]=min_ind;
size++;
tot=mincost;
if(size>=n){
    break;
}
}
cout<<"\nMinimum Total Cost is :- " << tot << endl;
cout<<"\nPath :- ";
for(int i=0; i<size; i++){
    cout<<visited[i]<<" -> ";
}
}

```