```
#include<bits/stdc++.h>
#include<vector>
using namespace std;

int n;
int mat[20][20];

vector<int>cost;

vector<vector<int> > vect;

int inf=999;
int start=0;

int shortest_path_sum()
{

    vector<int> nodes;
    for(int i=0;i<n;i++)
    {
        if(i != start)
        {
            nodes.push_back(i);
        }
    }

    int n1 = nodes.size();

    int shortest_path = INT_MAX;

    while(next_permutation(nodes.begin(),nodes.end()))

    {

        vector<int>temp;

                int path_weight = 0;

        int j = start;
        temp.push_back(j);

        for (int i = 0; i < n; i++)
        {
            path_weight += mat[j][nodes[i]];
            j = nodes[i];
            temp.push_back(j);
        }
        path_weight += mat[j][start];

        vect.push_back(temp);

        cost.push_back(path_weight);

        shortest_path = min(shortest_path, path_weight);
    }

        return shortest_path;
}
```

```
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];
		}
	}
	for(int i=0; i<n; i++){
		for(int j=0; j<n; j++){
			cout<<mat[i][j]<<"  ";
		}
		cout<<"\n\n";
	}
	cout<<"Min Cost "<<shortest_path_sum()<<endl;
	cout<<"\n";
	for(int i=0; i<vect.size(); i++){
		for(int j=0; j<vect[i].size(); j++){
			cout<<vect[i][j]<<"  ";
		}
		cout<<cost[i];
		cout<<"\n\n";
	}
	return 0;
}
```