Code:

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

#define INF 99999

int pi[100][100], dist[100][100];

void printPath(int i, int j){
  if(i==j){
    cout<<i<<" ";
    return;
  }
  else if(j == -1){
    cout<<"No Path";
  }
  else{
    printPath(i, pi[i][j]);
    cout<<j<<" ";
  }
}

void floyd(int n) {
  for (int k = 0; k < n; k++) {
    for (int i = 0; i < n; i++) {
      for (int j = 0; j < n; j++) {
        if (dist[i][k] != INF && dist[k][j] != INF &&
            dist[i][j] > (dist[i][k] + dist[k][j])) {
          dist[i][j] = dist[i][k] + dist[k][j];
          pi[i][j] = pi[k][j];
        }
      }
    }
  }
}

void printMatrix(int dist[100][100], int pi[100][100], int n) {
  cout << "\nDistance matrix" << endl;
  for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
      cout << dist[i][j] << " ";
    }
    cout << endl;
  }
  cout << "\nPI matrix" << endl;
  for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
      cout << pi[i][j] << " ";
    }
    cout << endl;
  }
}

int main() {

  int n;

  cout << "Enter the vertices: " << endl;
  cin >> n;

  cout << "\nEnter the adjacency matrix: " << endl;
  for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
      cin >> dist[i][j];
      if (i == j || dist[i][j] == INF)
        pi[i][j] = -1;
      else
        pi[i][j] = i;
    }
  }

  floyd(n);
  printMatrix(dist, pi, n);

  cout<<"\nAll paths are: "<<endl;
  for(int i=0; i<n; i++){
    for(int j=0; j<n; j++){
      cout<<"\nPath between "<< i <<" - "<< j << " ";
      printPath(i,j);
    }
    cout<<endl;
  }

  return 0;
}
```

Sample Output:

```
Enter the vertices:
3
Enter the adjacency matrix:
0 4 11
6 0 2
3 99999 0
Distance matrix
0 4 6
5 0 2
3 7 0

PI matrix
-1 0 1
2 -1 1
2 0 -1

All paths are:
Path between 0 - 0 0
Path between 0 - 1 0 1
Path between 0 - 2 0 1 2

Path between 1 - 0 1 2 0
Path between 1 - 1 1
Path between 1 - 2 1 2

Path between 2 - 0 2 0
Path between 2 - 1 2 0 1
Path between 2 - 2 2
```