110) Warshalls algorithm

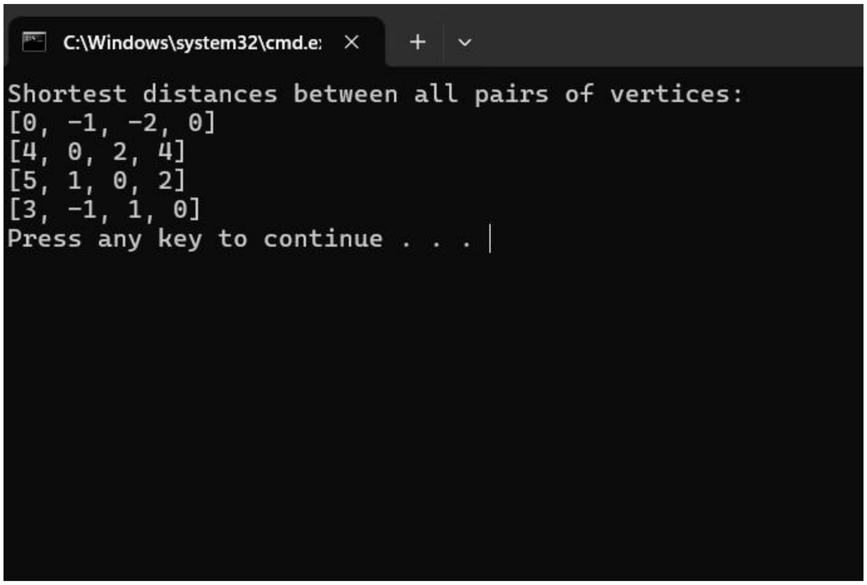
CODE:

```
import sys  def
floyd_warshall(graph, V):
    dist = [row[:] for row in graph]
         for k in range(V):         for i in range(V):
for j in range(V):                 if dist[i][j] > dist[i][k] +
dist[k][j]:                    dist[i][j] = dist[i][k] + dist[k][j]
         for i in range(V):         if dist[i][i] < 0:
print("Graph contains negative weight cycle")
return None
         return
dist
 if __name__ ==
"__main__":     graph = [
        [0, sys.maxsize, -2, sys.maxsize],
        [4, 0, 3, sys.maxsize],
        [sys.maxsize, sys.maxsize, 0, 2],
        [sys.maxsize, -1, sys.maxsize, 0]
    ]

    V = len(graph)

    shortest_paths = floyd_warshall(graph, V)
```

```
        if shortest_paths:         print("Shortest distances between all
pairs of vertices:")         for row in shortest_paths:
print(row)
```



```
Shortest distances between all pairs of vertices:
[0, -1, -2, 0]
[4, 0, 2, 4]
[5, 1, 0, 2]
[3, -1, 1, 0]
Press any key to continue . . .
```

TIME COMPLEXITY : $O(n^3)$