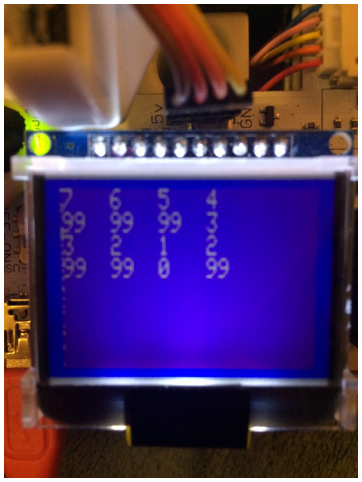
0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

Explain your algorithm in a write-up and show sample outputs of steps 1, 2 and 3.

Our algorithm accepts a 4x4 map with a specified goal node, and uses the cost function to compute minimum distances from all nodes to that goal. At each node, we check all adjacent non-visited nodes and store distances to find the optimal path. The distance accumulates by one for each node crossed as cost is either 1 for an open adjacent node or 99 if not open. The algorithm keeps track of the path, or parent of each node, by finding the minimum distance adjacent node and storing that index in a parents array for the current node. To find an optimal path, you give any starting node and it follows the parents of that node as it makes its way to the goal.

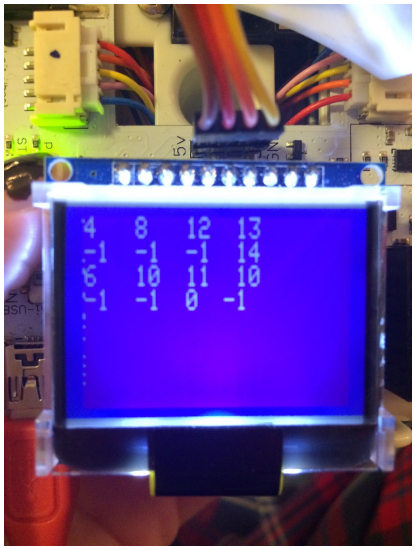
Step 1: *Use a simple 4x4 map as in the last exercise to validate that Dijkstra's returns the correct distances.*

The follow picture shows the distances that the Dijkstra's algorithm returns after exploring all the nodes in the map. The 99's correspond to the obstacles as seen in the map at the top of this page. Our starting point is at 11 (seen as 0 in the following image), and the ending point is 0 (which in the following image is a 7. The algorithm begins exploring at node 11 and fans out from there.



Step 2: *Modify the code so that Dijkstra's stores the id of the parent node.*

The following image shows the node number of the previously explored node. The ‘-1’ correspond to the obstacles in the map at the top of the write up.



Step 3: *Write code that returns a sequence of vertices from a given location on the map to the source.*

The image shows the node sequences that the robot will travel going from node 0 to node 11.