Assignment 8 (By Utkarsh Ranjan)

Problem 1

Variable

1. N => one byte, contains the number of node
2. ADJACENCY_MATRIX => The NxN matrix represented as an array of length NxN with the rows being concatenated.
3. Node => The source node

Algorithm

Dijkstra's Algorithm

- We maintain two sets, one set contains vertices included in the shortest-path tree, other set includes vertices not yet included in the shortest-path tree.
- At every step of the algorithm, we find a vertex that is in the other set (set of not yet included) and has a minimum distance from the source.

```
function Dijkstra(Graph, source):

    for each vertex v in Graph.Vertices:
        dist[v] ← INFINITY
        add v to Q
    dist[source] ← 0

    while Q is not empty:
        u ← vertex in Q with min dist[u]
        remove u from Q

        for each neighbor v of u still in Q:

            alt ← dist[u] + Graph.Edges(u, v)
            if alt < dist[v]:
                dist[v] ← alt

    return dist[]
```

Design of the program

a. Copying variable

Register	Address	Value Stored
R3	0x30000000	N
R4	0x40000000	Source Node
R6	0x60000000	Adjacency Matrix
R7	0x70000000	Shortest Distance Vector
R9	0x90000000	Q

N, Source Node and Adjacency Matrix are first copied into the following register.
The Shortest Distance vector and Q (A bool array) were initialized further in the programme.

```
FOR_COPY
        LDRB    R0,[R2],#01         ; copying the adjacency list to 0x60000000
        STRB    R0,[R6],#01
        SUBS    R8,R8,#01
        CMP     R8,#0
        BNE     FOR_COPY
        ; R8,R1,R0,R2 are free

        MOV     R1,R7
FOR_INIT
        LDR     R0,=0xFF            ; storing infinity as initial distance form
the source
        STRB    R0,[R1],#01
        MOV     R9,#01              ; Adding all vertex to the array Q as non-
visited
        ADD     R9,#01
        SUBS    R5,R5,#01
        CMP     R5,#0
        BNE     FOR_INIT
        LDR     R0,=0x00            ; updating the value of dist(source) in this
array to 0
        LDR     R2,[R4]
        STRB    R0,[R7,R2]
        ; R0,R8,R2,R5,R1 are free
```

b. Main Loop

- The programme has a main loop (similar to the pseudo code mentioned in this report)
- This loop further has 3 more loops.

c. Check Loop

- This loop covers " while Q is not empty:" of the pseudo code.
- It checks if the array stored at location in the register R9 contains an element which is 1. If all are 0 means every nodes are visited and Q is empty so the programme terminates.

d. First Loop

- This loop covers following part of the pseudo code

```
u ← vertex in Q with min dist[u]
remove u from Q
```

e. Second Loop

- This loop covers this part of the pseudo code

```
 for each neighbor v of u still in Q:
        alt ← dist[u] + Graph.Edges(u, v)
        if alt < dist[v]:
              dist[v] ← alt
```