**Lecture 6-7 Assignments Road Network**

1. a.) Assigning the array true from 1st [0] and 3rd position [2] and since bools of arrays [8] the following undeclared in the array is False

```c
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))

int main(){


    bool pathway[8] = {[0] = true, [2] = true};

    for(int i = 0; i < NUM_PATHWAYS; i++){

        if (pathway[i]){
            printf("pathway[&d] is open \n", i);
        }
        else {
            printf("pathway[&d] is close \n", i);
        }
    }
    return 0;
}
```

```
C:\Users\Hp\Documents\1_a.exe
pathway[&d] is open
pathway[&d] is close
pathway[&d] is open
pathway[&d] is close
pathway[&d] is close
pathway[&d] is close
pathway[&d] is close
pathway[&d] is close

Process returned 0 (0x0)   execution time : 7.042 s
Press any key to continue.
```

1. B.) From the arrays since the first three is only declared “true, false, true” therefore the next succeeding undeclared will be read is false

```c
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))

int main(){


    bool pathway[8] = {true, false, true};

    for(int i = 0; i < NUM_PATHWAYS; i++){

        if (pathway[i]){
            printf("pathway[&d] is open \n", i);
        }
        else {
            printf("pathway[&d] is close \n", i);
        }
    }
    return 0;
}
```

```
"C:\Users\Hp\Documents\1 letter b.exe"
pathway[&d] is open
pathway[&d] is close
pathway[&d] is open
pathway[&d] is close
pathway[&d] is close
pathway[&d] is close
pathway[&d] is close
pathway[&d] is close

Process returned 0 (0x0)   execution time : 0.017 s
Press any key to continue.
```

**Lecture 6-7 Assignments Road Network**

2. First upon creating the diagram of adjacency creating of the column as label or assignation from the column using for loop of char.

```c
#include<stdbool.h>

#define SIZE 8

int main(){
    //Initialize road networks
    bool road_networks[SIZE][SIZE] = {
        {1, 1, 0, 0, 0, 1, 0, 0},
        {1, 1, 1, 0, 0, 0, 0, 0},
        {0, 1, 1, 0, 1, 1, 0, 0},
        {0, 0, 0, 1, 1, 0, 0, 0},
        {0, 0, 0, 1, 1, 0, 0, 0},
        {1, 0, 1, 0, 0, 1, 0, 0},
        {1, 0, 0, 1, 0, 0, 1, 0},
        {0, 0, 0, 0, 0, 1, 0, 1}
    };

    // Column
    char col[8] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    for (int i = 0; i < 8; i++){
        if (i < 2 || i > 3){
            printf("      %c", col[i]);
        }
        else if (i == 2 || i == 3){
            printf("    [%c]", col[i]);
        }
    }
        printf("\nA     ");


    return 0;
}
```

```
C:\Users\Hp\Documents\wehh.exe
      A       B     [C]     [D]       E       F       G       H
A
Process returned 0 (0x0)   execution time : 2.922 s
Press any key to continue.
```

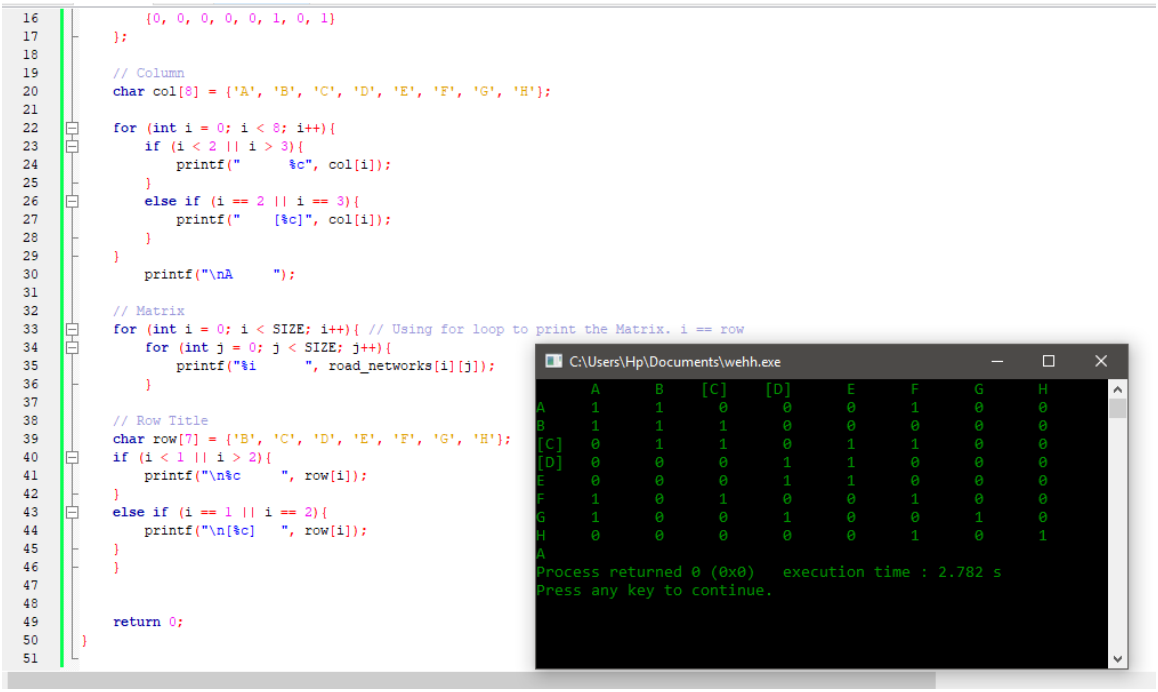
Now inserting the rows of the arrays using nested loop to manipulate each tile of the diagram road network. The node from rows and column (matrix).

```c
        {1, 0, 0, 1, 0, 0, 1, 0},
        {0, 0, 0, 0, 0, 1, 0, 1}
    };

    // Column
    char col[8] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    for (int i = 0; i < 8; i++){
        if (i < 2 || i > 3){
            printf("      %c", col[i]);
        }
        else if (i == 2 || i == 3){
            printf("    [%c]", col[i]);
        }
    }
        printf("\nA     ");

    // Matrix
    for (int i = 0; i < SIZE; i++){ // Using for loop to print the Matrix. i == row
        for (int j = 0; j < SIZE; j++){
            printf("%i       ", road_networks[i][j]);
        }


    return 0;
}
}
```

```
C:\Users\Hp\Documents\wehh.exe
      A       B     [C]     [D]       E       F       G       H
A     1       1       0       0       0       1       0       0
Process returned 0 (0x0)   execution time : 3.059 s
Press any key to continue.
```

**Lecture 6-7 Assignments Road Network**

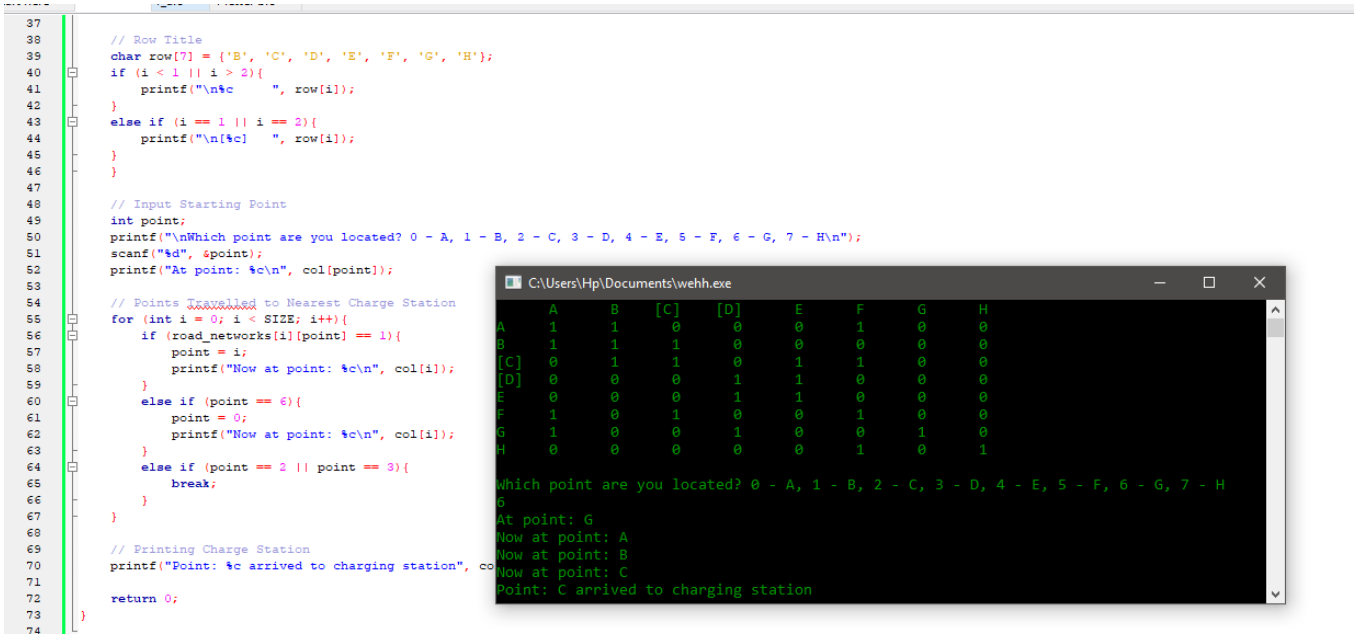
From the matrixes each assigned from the arrays will now display all the rows containing each tiles respective value to display all via for loop all elements in matrixes displayed



```
        {0, 0, 0, 0, 0, 1, 0, 1}
    };

    // Column
    char col[8] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

    for (int i = 0; i < 8; i++){
        if (i < 2 || i > 3){
            printf("      %c", col[i]);
        }
        else if (i == 2 || i == 3){
            printf("    [%c]", col[i]);
        }
    }
        printf("\nA     ");

    // Matrix
    for (int i = 0; i < SIZE; i++){ // Using for loop to print the Matrix. i == row
        for (int j = 0; j < SIZE; j++){
            printf("%i      ", road_networks[i][j]);
        }

    // Row Title
    char row[7] = {'B', 'C', 'D', 'E', 'F', 'G', 'H'};
    if (i < 1 || i > 2){
        printf("\n%c     ", row[i]);
    }
    else if (i == 1 || i == 2){
        printf("\n[%c]   ", row[i]);
    }
    }


    return 0;
}
```

```
C:\Users\Hp\Documents\wehh.exe
      A      B     [C]    [D]      E      F      G      H
A     1      1      0      0      0      1      0      0
B     1      1      1      0      0      0      0      0
[C]   0      1      1      0      1      1      0      0
[D]   0      0      0      1      1      0      0      0
E     0      0      0      1      1      0      0      0
F     1      0      1      0      0      1      0      0
G     1      0      0      1      0      0      1      0
H     0      0      0      0      0      1      0      1
A
Process returned 0 (0x0)   execution time : 2.782 s
Press any key to continue.
```

Now since the path is always = 1 skipping the conditional this method will fail when a 1 is found from the chosen point and terminating the charging station if it equals from 1 and 2. Final value terminates from 1 in the row displays. If the station point at 2 and 3 the program will display the number of charging station before terminating all.



```
    // Row Title
    char row[7] = {'B', 'C', 'D', 'E', 'F', 'G', 'H'};
    if (i < 1 || i > 2){
        printf("\n%c     ", row[i]);
    }
    else if (i == 1 || i == 2){
        printf("\n[%c]   ", row[i]);
    }
    }

    // Input Starting Point
    int point;
    printf("\nWhich point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H\n");
    scanf("%d", &point);
    printf("At point: %c\n", col[point]);

    // Points Travelled to Nearest Charge Station
    for (int i = 0; i < SIZE; i++){
        if (road_networks[i][point] == 1){
            point = i;
            printf("Now at point: %c\n", col[i]);
        }
        else if (point == 6){
            point = 0;
            printf("Now at point: %c\n", col[i]);
        }
        else if (point == 2 || point == 3){
            break;
        }
    }

    // Printing Charge Station
    printf("Point: %c arrived to charging station", co

    return 0;
}
```

```
C:\Users\Hp\Documents\wehh.exe
      A      B     [C]    [D]      E      F      G      H
A     1      1      0      0      0      1      0      0
B     1      1      1      0      0      0      0      0
[C]   0      1      1      0      1      1      0      0
[D]   0      0      0      1      1      0      0      0
E     0      0      0      1      1      0      0      0
F     1      0      1      0      0      1      0      0
G     1      0      0      1      0      0      1      0
H     0      0      0      0      0      1      0      1

Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
6
At point: G
Now at point: A
Now at point: B
Now at point: C
Point: C arrived to charging station
```

**Lecture 6-7 Assignments Road Network**

References:

*Adjacency Matrix in C | Adjacency matrix representation of graph*. (n.d.). EQuestionAnswers. Retrieved May 12, 2022, from https://www.equestionanswers.com/c/c-adjacency-matrix.php

*C program to implement Adjacency Matrix of a given Graph*. (2020, May 21). GeeksforGeeks. Retrieved May 12, 2022, from https://www.geeksforgeeks.org/c-program-to-implement-adjacency-matrix-of-a-given-graph/

Github link: https://github.com/elipancarl/CMSC21.git