

CMSC 21- Lesson 6 & 7 Assignment

1. a. Revise line 16 such that you use a designated initializer to set pathways 0 and 2 to true, and the rest will be false. Make the initializer as short as possible.

```
12
13      Only pathways 0 and 3 are open for transportation. The rest are close.
14
15      */
16
17      bool pathway[8]={0} =true, [2]= true};
18
19      for(int i= 0; i < NUM_PATHWAYS; i++){
20          /*
21
22          Display the status of each pathway.
23
24          Remember that pathway type is bool so its elements are either true/false - 1/0.
25
26          */
27
28          if (pathway[i]){
29              printf("pathway [%d] is open \n", i);
30          }
31          else{
32              printf("pathway [%d] is close \n", i);
33          }
34      }
35      return 0;
36  }
37
```

- b. Revise line 16 such that the initializer will be short as possible (without using a designated initializer)

```
12
13      Only pathways 0 and 3 are open for transportation. The rest are close.
14
15      */
16
17      bool pathway[8]={1,0,1};
18
19      for(int i= 0; i < NUM_PATHWAYS; i++){
20          /*
21
22          Display the status of each pathway.
23
24          Remember that pathway type is bool so its elements are either true/false - 1/0.
25
26          */
27
28          if (pathway[i]){
29              printf("pathway [%d] is open \n", i);
30          }
31          else{
32              printf("pathway [%d] is close \n", i);
33          }
34      }
35      return 0;
36  }
37
```

2.

The programming assignment was more challenging compared to the previous assignments as it required deeper understanding of the topics involving loops and arrays. For determining the size of the 2d array I used macros as hinted on the activity guide. I then initialized two almost identical arrays, one for the 2d array which will have brackets on letter C and D and one for printing the output which does not have a bracket on any of the letters. I then manually initialized the content of the 2d array which is 8 x 8 in size. In printing the 2d array, I first printed the letters on the upper part of the 2d array and then used a for loop that iterates through the rows and printing the corresponding letter per row. I then nested another loop which also prints the contents of the 2d array per row. After that I printed a statement that guides the user to encode an input which will be the starting point of the program. I then declared a locator variable which will be used for traversing through the 2d array. In the loop, there is a conditional that prints the location when it lands on a true value. Inside that conditional, there is a nested set of conditionals that checks if the column is located on either the C or D column. If the condition on either nested conditional is met the loop will be ended. If the locator reaches the value of 7, the loop will end on the next iteration. I used a conditional that resets the value of the locator back to 0 so that the loop will continue until it reaches a charging station. Lastly, an else statement increments a value of 1 to the locator variable when the program lands on a false value on the 2d array.

The activity is very challenging as it requires a deep understanding on loops and multidimensional arrays. The printing of the 2d array quite took sometime due to difficulties in the desired display. The main loop for finding the nearest charging station also quite took some time due to debugging algorithms that result in infinite loops. The problem with my solution is that it rarely goes to charging station D and always go to charging station C despite being farther away from the starting point. The instances that the program ends the loop in D is when the starting point is D and E. Nevertheless, the program ends the loop on the intended charging stations.

```
Start here X as1b.c X as2.c X
1 #include <stdio.h>
2
3 #define N_ROWS 8 //using macros to define array size
4 #define N_COLUMNS 8
5
6 int main(void) { //declaring the main function
7     int user_input, i, j; //declaring variable for user input and for loop initialization
8     char *road_stations[]={"A", "B", "C", "D", "E", "F", "G", "H"}; //initializing array of strings for final output
9     char *array_stations[]={" A", " B", "[C]", "[D]", " E", " F", " G", " H"}; //initializing array of strings for 2d matrix
10
11     int road_networks[N_ROWS][N_COLUMNS]= { //initializing 2d array
12         { 1, 1, 0, 0, 0, 1, 0, 0},
13         { 1, 1, 1, 0, 0, 0, 0, 0},
14         { 0, 1, 1, 0, 1, 1, 0, 0},
15         { 0, 0, 0, 1, 1, 0, 0, 0},
16         { 0, 0, 0, 1, 1, 0, 0, 0},
17         { 1, 0, 1, 0, 0, 1, 0, 0},
18         { 1, 0, 0, 1, 0, 0, 1, 0},
19         { 0, 0, 0, 0, 0, 1, 0, 1}
20     };
21
22     printf("      A      B      [C]      [D]      E      F      G      H\n"); //manually printing the header part of the 2d array
23     for(i= 0; i < N_ROWS; i++){ //iteration through array rows
24         printf("%s", array_stations[i]); //printing row labels on the side
25         for (j= 0; j < N_COLUMNS; j++){ //iteration through the columns
26             printf("%8d", road_networks[i][j]); //printing 1 and 0 values on the 2d array
```

```
Start here X as1b.c X as2.c X
33 printf("Which point are you located? 0- A, 1- B, 2- C, 3- D, 4- E, 5- F, 6- G, 7- H \n"); //taking an input from the user for station
34 scanf("%d", &user_input); //storing user input in previously declared variable
35
36 printf("\nAt point: %s", road_stations[user_input]); //printing starting point
37
38 int locator; //declaring locator variable, to navigate through the 2d array
39
40 for(locator= 0; locator < N_COLUMNS;){ //iteration of locator through the columns of the 2d array
41     if(road_networks[user_input][locator] == 1){ //if statement when the iteration arrives at a true value
42         printf("\nNow at point: %s", road_stations[locator]); //printing the current location
43
44         if (locator == 2){ //when locator arrives at column C
45             printf("\nArrived at charging station: %s", road_stations[locator]); //printing the indication that the iteration has reached
46             break; //ending the for loop
47         }
48         else if (locator == 3){ //when locator arrives at column D
49             printf("\nArrived at charging station: %s", road_stations[locator]); //printing the indication that the iteration has reached
50             break; //ending the for loop
51         }
52         user_input = locator; //setting the value for the default value of the row (user input) into the value of the locator
53         locator += 1; //incrementing 1 to the value of the locator to avoid an infinite loop when arriving at station A
54     }
55     else if (locator == 7){ //if statement when the locator reaches the value 7 without arriving at the nearest charging station
56         locator = 0; //resetting the value of the locator back to 0 in order to continue the loop
57     }
58     else{
59         locator += 1; //loop will increment the value of 1 to the locator when it lands on a false location
60     }
61 }
```