

1.

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

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

2.

```
1  #include <stdio.h>
2
3  /*using macros to define size of the array*/
4  #define N_ROWS 8
5  #define N_COLUMNS 8
6
7  int main(void){
8      /*declaring variables*/
9      int user_input, i, j;
10
11     /*array of strings for final output*/
12     char *road_station[]={ "A", "B", "C", "D", "E", "F", "G", "H"};
13
14     /*array for 2d matrix printing*/
15     char *array_station[]= { "  A", "  B", "[C]", "[D]", "  E", "  F", "  G", "  H"};
16
17     /*2d array*/
18
19     int road_networks[N_ROWS][N_COLUMNS]= {
20         { 1, 1,0,0, 0, 1, 0, 0},
21         { 1, 1,1,0, 0, 0, 0, 0},
22         { 0, 1,1,0, 1, 1, 0, 0},
23         { 0, 0,0,1, 1, 0, 0, 0},
24         { 0, 0,0,1, 1, 0, 0, 0},
25         { 1, 0,1,0, 0, 1, 0, 0},
26         { 1, 0,0,1, 0, 0, 1, 0},
27         { 0, 0,0,0, 0, 1, 0, 1}
28     };
29
30
31     /*array header*/
32     printf("      A      B      [C]      [D]      E      F      G      H\n");
33     for(i= 0; i < N_ROWS; i++){ //iterate through rows
34         printf("%s", array_station[i]);
35         for (j= 0; j < N_COLUMNS; j++){ //iterate through columns
36             printf("%8d", road_networks[i][j]); //print 1 and 0 values
37         }
38         printf("\n");
39     }
40
41     printf("\n");
```

```

40
41 printf("\n");
42
43 /*asking input from the user*/
44 printf("Which point are you located? 0- A,1- B, 2- C, 3- D, 4- E, 5- F, 6- G, 7- H \n");
45 scanf("%d",&user_input);
46
47 /*print starting point*/
48 printf("\nAt point: %s", road_station[user_input]);
49
50 /*declare locator variables*/
51 int locator;
52
53 /*iterate Locator through columns*/
54 for(locator= 0; locator < N_COLUMNS;){
55     if(road_networks[user_input][locator] == 1){ //true value
56         printf("\nNow at point: %s",road_station[locator]); //print current location
57
58         if (locator == 2){ // when the locator is at column C
59             printf("\nArrived at charging station: %s", road_station[locator]); //print indication that the iteration has arrived at charging station C
60             break;
61         }
62         else if (locator == 3){ //when locator arrives at column D
63             printf("\nArrived at charging station: %s", road_station[locator]); //print the indication that the iteration has arrived at charging station D
64             break;
65         }
66         user_input = locator; //setting the value of the user input into the value of the locator
67         locator += 1; //increment 1 to the value of the locator to avoid an infinite loop
68     }
69     else if(locator == 7){ //when the locator reaches the value 7 without arriving at the nearest charging station
70         locator = 0; //reset value of the locator
71     }
72     else{
73         locator += 1; //increment the value of 1 when the locator lands on a false location
74     }
75 }
76 }
77 return 0;
78 }

```

First, we need to define the size of the array using macros. We determine the size by rows and columns. Then we declare variables such as the user input, i and j as integers. i for rows and j for columns. After that, we initialize the arrays for the stations. The elements would be in char data type. Then we create a multidimensional array using the values of 1 and 0 from the given table. After initializing the arrays, we are going to display the matrix on the console using the for loop. After that, we take input from the user and print the starting point. using the locator, we will iterate through the columns and if the locator lands on a true value, then the console will print the location. However, if the locator is at column C and D, the console will display that the locator arrived at the charging points. We also set the value of the user input as the locator and increment it by 1 to avoid infinite loop. Lastly, if the locator reaches the value of 7 without arriving at a station, the value of the locator resets back to 0,

At first, I was having a hard time determining the correct algorithm. After sometime, I finally got the working algorithm by reviewing the lectures. I realized that it is tricky to program multidimensional arrays if you don't have a deep understanding about it.

GitHub Link:

<https://github.com/purseerus/CMSC-21/tree/main/Lecture%206-7/Assignments>