Ellabelle E. Garcia
BS in Computer Science - I
Ara Abigail E. Ambita
CMSC 21

CMSC 21

Lectures 6-7 Assignment

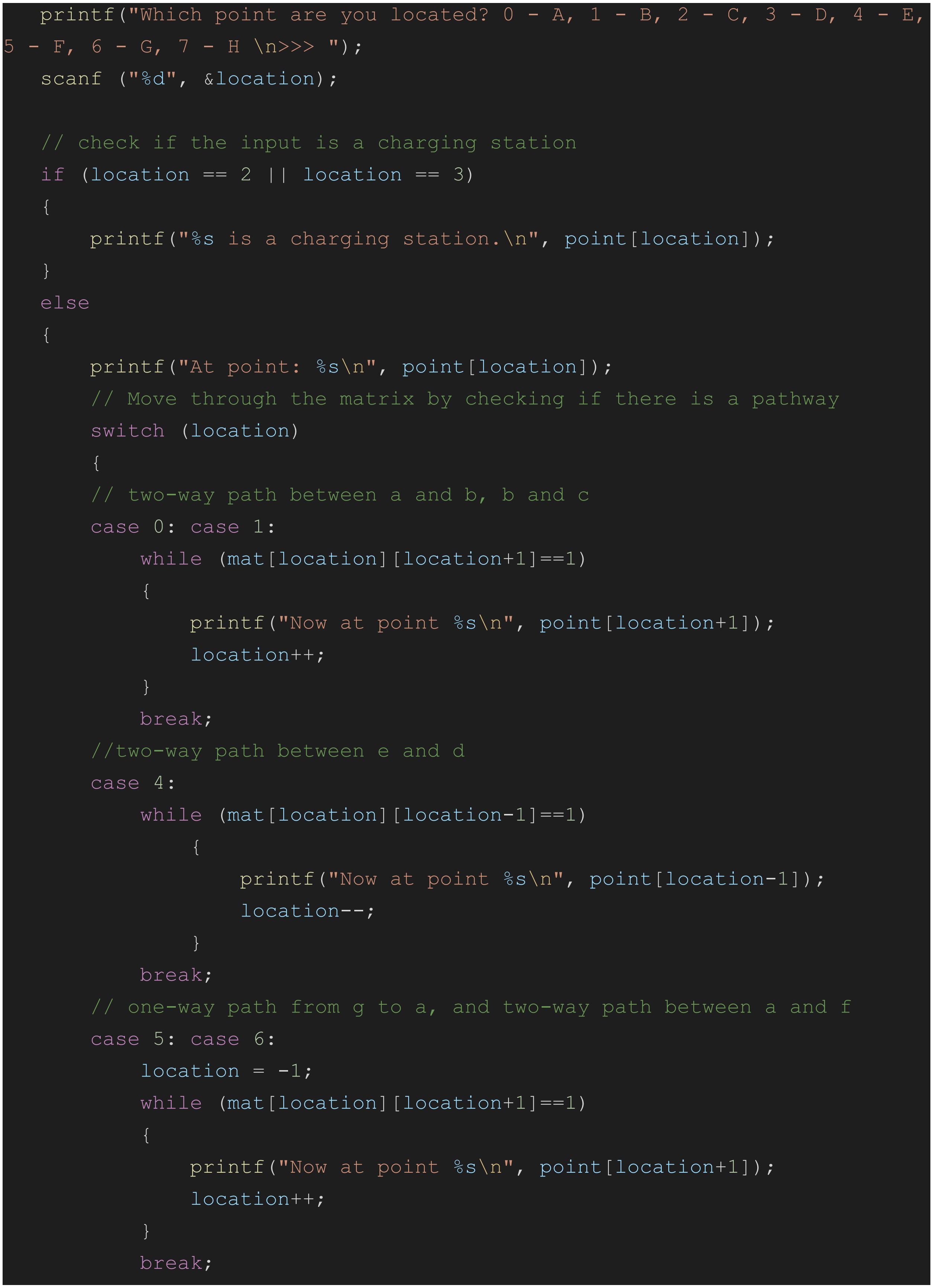
Link: https://github.com/ellabellegarcia/CMSC21/tree/main/Lecture6-7/Assignments

As a programming assignment:

1. Declare and initialize a road_networks multidimensional array that represents the adjacency matrix
2. Display the adjacency matrix. Put a bracket to the points/destinations that are considered as charging stations, e.g. [c], [d]
3. Given a point / destination, determine the nearest charging station. For example, if you are in point a, the nearest charging station is point c. If you are in point e, the nearest charging station is point d.
4. Bonus: Use a macro to define the size of the 2d array

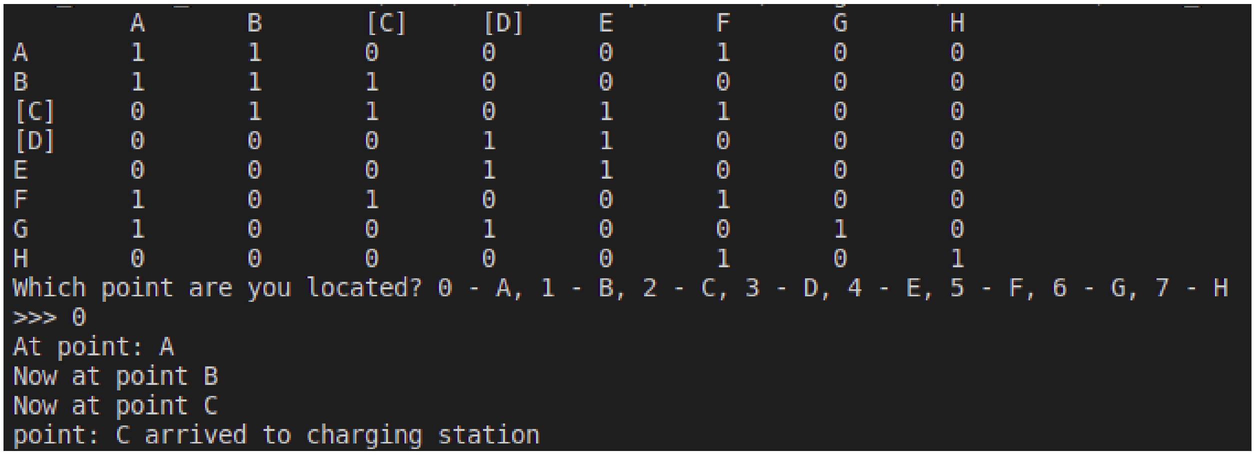
Code:

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

#define MAT_SIZE 8
int i, j, location;

// Function for printing the matrix
void printMatrix(char station[MAT_SIZE][MAT_SIZE], bool
mat[MAT_SIZE][MAT_SIZE])
{
   printf("\tA\tB\t[C]\t[D]\tE\tF\tG\tH\n");
   for (i = 0; i < MAT_SIZE ; i++) {
       printf("%s", station[i]);
       for (j = 0; j < MAT_SIZE ; j++)
           printf("\t%d", mat[i][j]);
       printf("\n");
   }
}

// Function for printing the points/location
void printPoints(char *point[MAT_SIZE],bool mat[MAT_SIZE][MAT_SIZE]){
```

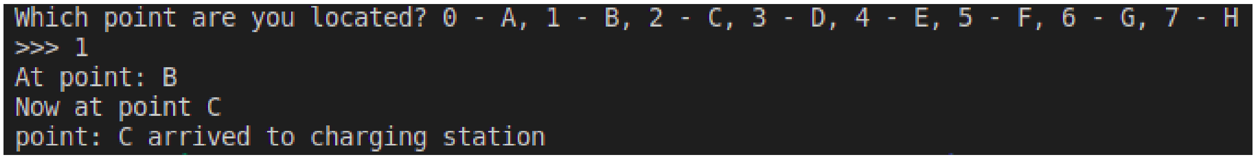


```
    printf("Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H \n>>> ");
    scanf ("%d", &location);

    // check if the input is a charging station
    if (location == 2 || location == 3)
    {
        printf("%s is a charging station.\n", point[location]);
    }
    else
    {
        printf("At point: %s\n", point[location]);
        // Move through the matrix by checking if there is a pathway
        switch (location)
        {
        // two-way path between a and b, b and c
        case 0: case 1:
            while (mat[location][location+1]==1)
            {
                printf("Now at point %s\n", point[location+1]);
                location++;
            }
            break;
        //two-way path between e and d
        case 4:
            while (mat[location][location-1]==1)
                {
                    printf("Now at point %s\n", point[location-1]);
                    location--;
                }
            break;
        // one-way path from g to a, and two-way path between a and f
        case 5: case 6:
            location = -1;
            while (mat[location][location+1]==1)
            {
                printf("Now at point %s\n", point[location+1]);
                location++;
            }
            break;
        // No direct path
        default:
            printf("No Path Available!\n");
            break;
        }
        // Charging station
        if (location == 2 || location == 3)
        {
            printf("point: %s arrived to charging station\n",
point[location]);
        }
    }
}


// Driver Code
int main()
{
    char *point[] = {"A", "B", "C", "D", "E", "F", "G", "H"};
    char station[MAT_SIZE][MAT_SIZE] = {{"A  "}, {"B  "}, {"[C]"}, {"[D]"},
{"E  "}, {"F  "}, {"G  "}, {"H  "}};
    bool road_networks[MAT_SIZE][MAT_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},
    };

    printMatrix(station, road_networks);
    printPoints(point, road_networks);

    return 0;
}
```

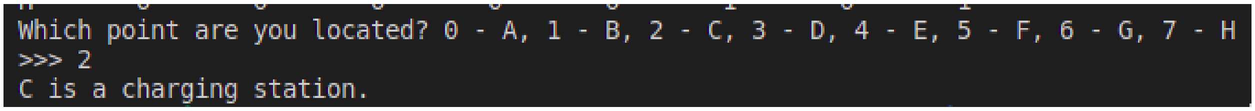
Output:



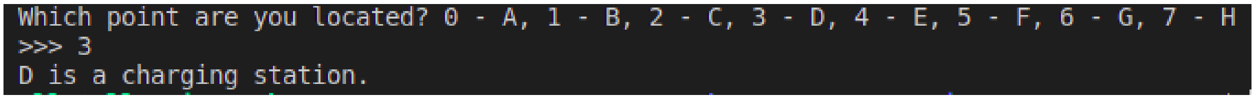
```
        A       B       [C]     [D]     E       F       G       H
A       1       1       0       0       0       1       0       0
B       1       1       1       0       0       0       0       0
[C]     0       1       1       0       1       1       0       0
[D]     0       0       0       1       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
>>> 0
At point: A
Now at point B
Now at point C
point: C arrived to charging station
```



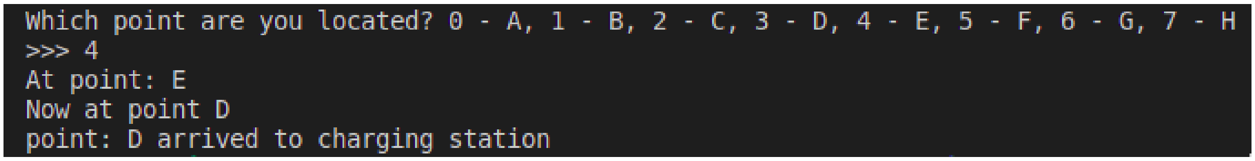
```
Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
>>> 1
At point: B
Now at point C
point: C arrived to charging station
```



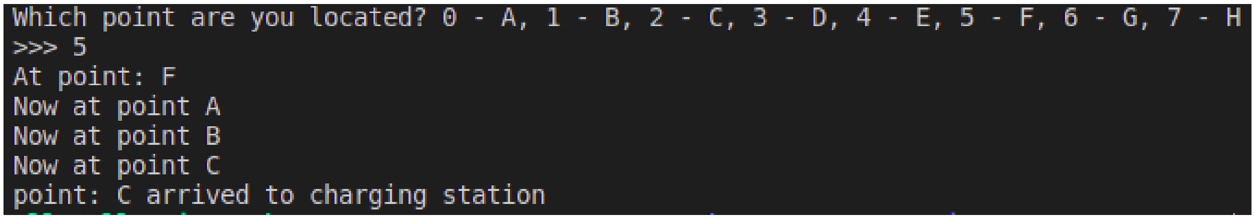
```
Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
>>> 2
C is a charging station.
```



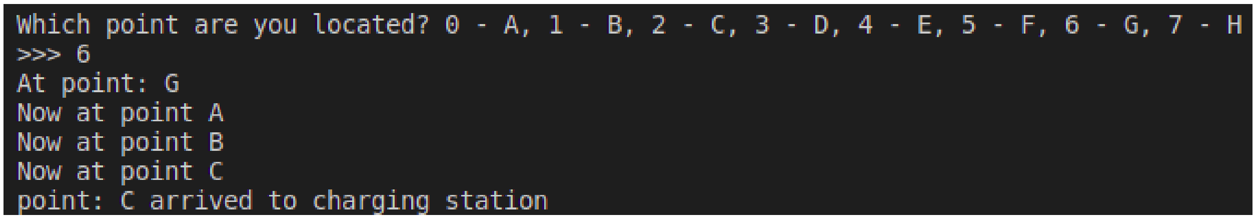
```
Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
>>> 3
D is a charging station.
```



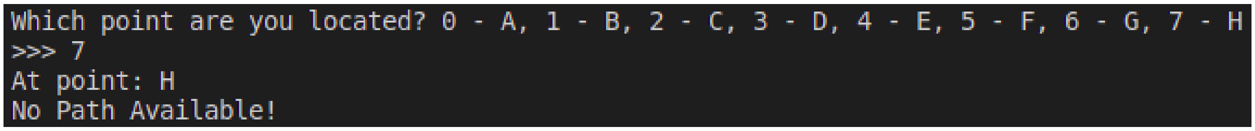
```
Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
>>> 4
At point: E
Now at point D
point: D arrived to charging station
```



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



```
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
```



```
Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
>>> 7
At point: H
No Path Available!
```

Discussion:

This program takes input from the user, asking which point they are located. It has three functions: for displaying the matrix and points and for declaring and initializing the arrays.

Matrix:

- The printMatrix function takes two parameters: the 2D array of the stations to be printed on the first column and the 2D array containing boolean values.
- Two nested for loops are needed to represent the rows and columns for printing the adjacency matrix.
- To maintain the format of the matrix, a new line must be printed at the end of each row.

Points/Location:

- The printPoints function takes two parameters: the array of letters representing the points and the 2D array containing boolean values.
- In this function, the user will be asked which point they are located.
- The user input will be taken as a condition in the if-else statement and switch-case statements.
- If the user input is equal to 2 or 3, it means that the user is already in the charging station. A statement will be printed, prompting the user about it.
- Otherwise, the user will have to move to another point until the user reaches the charging station.
- The indexes for the row and column are utilized to move through the matrix.
- Some points have one-way, two-way, and no direct paths, so I used a switch-case statement to set up the varying conditions.
- The user will also be notified if they arrive at the charging station.

Main function:

- In the main function, three arrays are initialized.
  - The point array contains the letters to be used in printing the current location of the user.
  - The stations which are displayed in the first column of the adjacency matrix are stored in the station array.
  - The road_networks is an adjacency matrix of booleans representing the points/destinations.
- To display the adjacency matrix and the user's current location, the printMatrix, and printPoints functions are called.

.