

Chester Ken Gallego

## Lecture 6-7 Assignment

1. A.

```
// Set pathways 0 and 2 to true using a designated initializer  
bool pathway[8] = {[0] = true, [2] = true};
```

B.

```
// Set pathways 0 and 2 to true without using a designated initializer  
bool pathway[8] = {true, false, true};
```

2. A

```
// Initialize the arrays  
int 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},  
};  
char points[SIZE] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};
```

First, set SIZE to 8 using the #define preprocessor. Then, using the keyword SIZE, initialize the arrays. Since the road\_networks array is a 2D array with the same number of rows and columns, simply use the SIZE keyword twice. Since there are eight points, use the SIZE keyword again to create an array of size eight for characters.

```

// Display the points as column titles
for (int i = 0; i < SIZE; i++)
{
    if (i == 0)
    {
        printf("\t");
    }
    if (i == 2 || i == 3)
    {
        printf("[%c]\t", points[i]);
    }
    else
    {
        printf("%c\t", points[i]);
    }
}
// Print a newline to go to the next row
printf("\n");

```

Then, display the points as column titles. Remember that these points are stored in the points array as characters. Iterate through the array using a for loop and print the eight points. On the first iteration, insert a Tab character to move point A to the second column of the table. Add brackets to charging stations.

```

// Display the rows and values of the 2D array
for (int i = 0; i < SIZE; i++)
{
    // Display the points as row titles
    if (i == 2 || i == 3)
    {
        printf("[%c]\t", points[i]);
    }
    else
    {
        printf("%c\t", points[i]);
    }
    // Display the values of the 2D array
    for (int j = 0; j < SIZE; j++)
    {
        printf("%d\t", road_networks[i][j]);
    }
    // Print a newline to move to the next row
    printf("\n");
}

```

Then, display the rows of the table. Display the points in the first column as row titles. Add brackets to charging stations. Then, print the value of each row of the array, ending with a newline to move on to the next row, and repeat until all rows have been displayed.

```
// Prompt the user to enter the point and display the corresponding letter
int point;
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", &point);
printf("At point: %c\n", points[point]);
```

After displaying the table representation of the 2D array, prompt the user to enter the point. Convert that point to its corresponding letter and display it.

```
// Locate the nearest charging point
for (int i = 0; point != 2 && point != 3; i++)
{
    // Determine whether the adjacent point is accessible
    if (road_networks[point][i])
    {
        // Set the current point to the nearest accessible adjacent point
        point = i;
        printf("Now at point: %c\n", points[point]);
    }
}
// Display a message if the user arrives at a charging point
printf("Point: %c arrived to charging station\n", points[point]);
```

Then, locate the nearest charging station in relation to the user's entry. If the user is in point A, B, C, F, or G, the nearest charging station is point C. If the user is in point D or E, the nearest charging station is point D. However, if the user is in point H, the program enters an infinite loop that eventually results to a segmentation fault because there is no route from point H to a charging station. Finally, if the program finds a way to a charging point, it will display a message.

B.

```
1 #include <stdio.h>
2
3 #define SIZE 8
4
5 // Declare the functions
6 void display_column(char points[SIZE]);
7 void display_row(char points[SIZE], int road_networks[SIZE][SIZE]);
8
9 int main(void)
10 {
11     // Initialize the arrays
12     int road_networks[SIZE][SIZE] = {
13         {1, 1, 0, 0, 0, 1, 0, 0},
14         {1, 1, 1, 0, 0, 0, 0, 0},
15         {0, 1, 1, 0, 1, 1, 0, 0},
16         {0, 0, 0, 1, 1, 0, 0, 0},
17         {0, 0, 0, 1, 1, 0, 0, 0},
18         {1, 0, 1, 0, 0, 1, 0, 0},
19         {1, 0, 0, 1, 0, 0, 1, 0},
20         {0, 0, 0, 0, 0, 1, 0, 1},
21     };
22     char points[SIZE] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};
23     // Call the functions
24     display_column(points);
25     display_row(points, road_networks);
26     // Prompt the user to enter the point and display the corresponding letter
27     int point;
28     printf("Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H\n");
29     scanf("%d", &point);
30     printf("At point: %c\n", points[point]);
31     // Locate the nearest charging point
32     for (int i = 0; point != 2 && point != 3; i++)
33     {
34         // Determine whether the adjacent point is accessible
35         if (road_networks[point][i])
36         {
37             // Set the current point to the nearest accessible adjacent point
38             point = i;
39             printf("Now at point: %c\n", points[point]);
40         }
41     }
42     // Display a message if the user arrives at a charging point
43     printf("Point: %c arrived to charging station\n", points[point]);
44     return 0;
45 }
46
47 void display_column(char points[SIZE])
48 {
49     // Display the points as column titles
50     for (int i = 0; i < SIZE; i++)
51     {
52         if (i == 0)
53         {
54             printf("\t");
55         }
56         if (i == 2 || i == 3)
57         {
58             printf("[%c]\t", points[i]);
59         }
60         else
61         {
62             printf("%c\t", points[i]);
63         }
64     }
65     // Print a newline to go to the next row
66     printf("\n");
67 }
68
69 void display_row(char points[SIZE], int road_networks[SIZE][SIZE])
70 {
71     // Display the rows and values of the 2D array
72     for (int i = 0; i < SIZE; i++)
73     {
74         // Display the points as row titles
75         if (i == 2 || i == 3)
76         {
77             printf("[%c]\t", points[i]);
78         }
79         else
80         {
81             printf("%c\t", points[i]);
82         }
83         // Display the values of the 2D array
84         for (int j = 0; j < SIZE; j++)
85         {
86             printf("%d\t", road_networks[i][j]);
87         }
88         // Print a newline to move to the next row
89         printf("\n");
90     }
91 }
92
```

Another version that uses functions.

GitHub: <https://github.com/chstrkn/CMSC21/tree/main/Lecture6-7/Assignments>