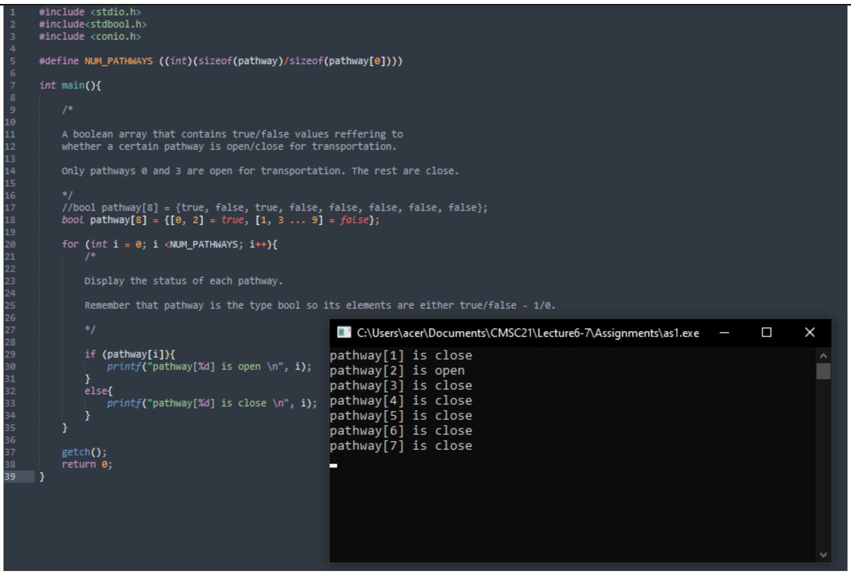
1

a



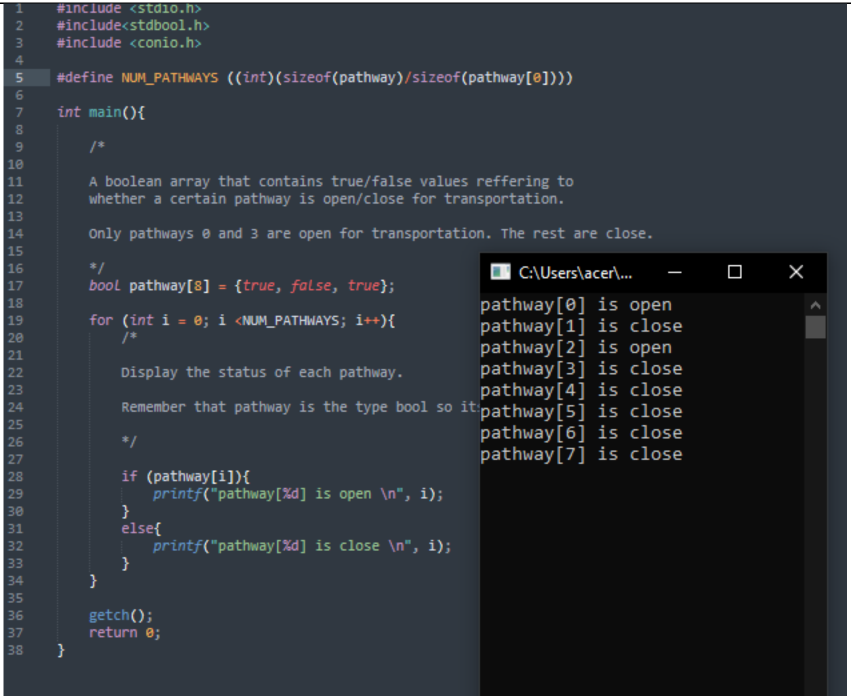
```
1   #include <stdio.h>
2   #include<stdbool.h>
3   #include <conio.h>
4
5   #define NUM_PATHWAYS ((int)(sizeof(pathway)/sizeof(pathway[0])))
6
7   int main(){
8
9       /*
10
11      A boolean array that contains true/false values reffering to
12      whether a certain pathway is open/close for transportation.
13
14      Only pathways 0 and 3 are open for transportation. The rest are close.
15
16      */
17      //bool pathway[8] = {true, false, true, false, false, false, false, false};
18      bool pathway[8] = {[0, 2] = true, [1, 3 ... 9] = false};
19
20      for (int i = 0; i <NUM_PATHWAYS; i++){
21          /*
22
23          Display the status of each pathway.
24
25          Remember that pathway is the type bool so its elements are either true/false - 1/0.
26
27          */
28
29          if (pathway[i]){
30              printf("pathway[%d] is open \n", i);
31          }
32          else{
33              printf("pathway[%d] is close \n", i);
34          }
35      }
36
37      getch();
38      return 0;
39  }
```

C:\Users\acer\Documents\CMSC21\Lecture6-7\Assignments\as1.exe

```
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

Revised line 16: bool pathway[8] = {[0, 2] = true};

b



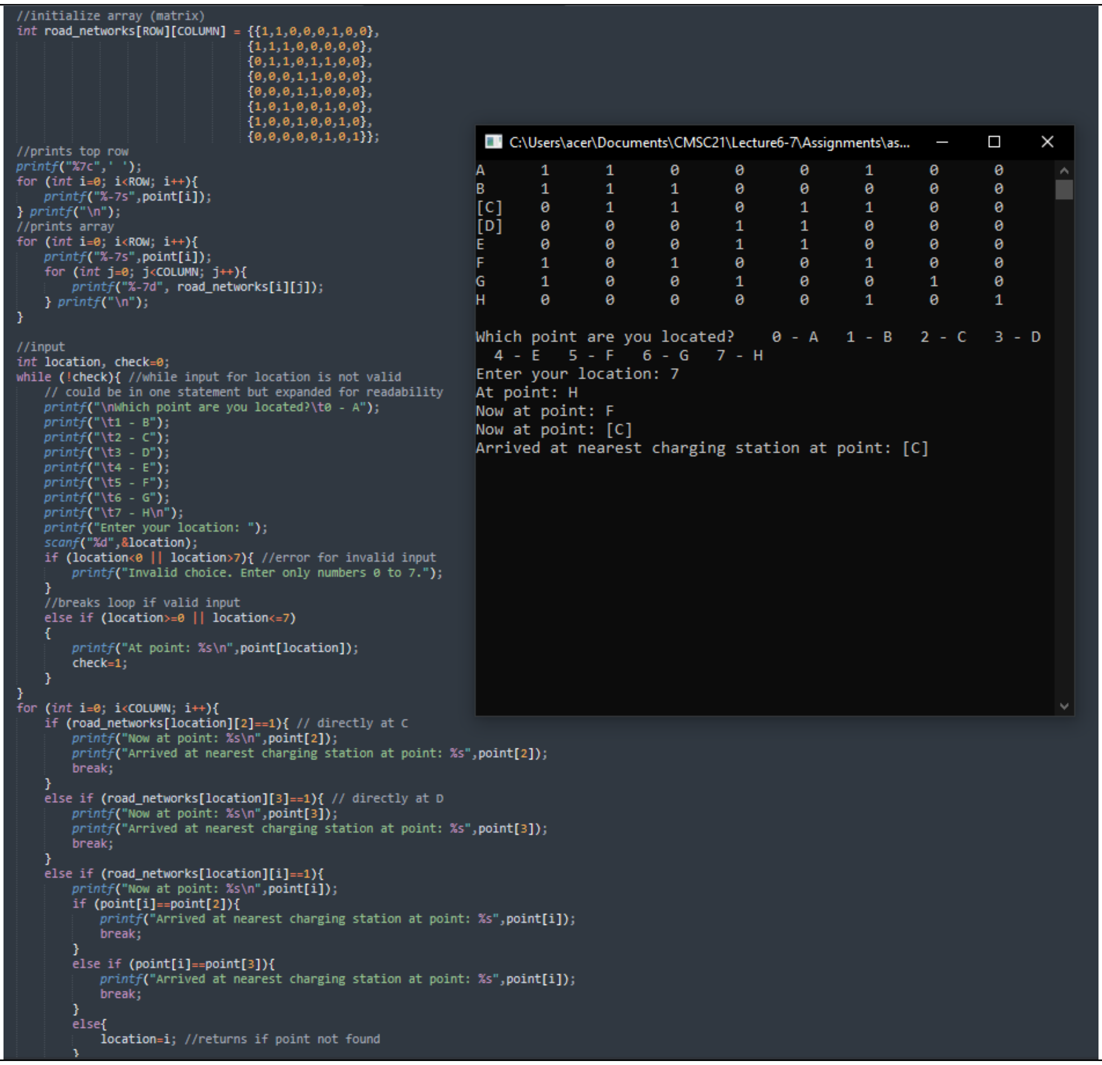
```
1   #include <stdio.h>
2   #include<stdbool.h>
3   #include <conio.h>
4
5   #define NUM_PATHWAYS ((int)(sizeof(pathway)/sizeof(pathway[0])))
6
7   int main(){
8
9       /*
10
11      A boolean array that contains true/false values reffering to
12      whether a certain pathway is open/close for transportation.
13
14      Only pathways 0 and 3 are open for transportation. The rest are close.
15
16      */
17      bool pathway[8] = {true, false, 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 is the type bool so it
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
36      getch();
37      return 0;
38  }
```

C:\Users\acer\...

```
pathway[0] is open
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

Revised line 16: bool pathway[8] = {true, false, true};

2



```c
//initialize array (matrix)
int road_networks[ROW][COLUMN] = {{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}};
//prints top row
printf("%7c",' ');
for (int i=0; i<ROW; i++){
    printf("%-7s",point[i]);
} printf("\n");
//prints array
for (int i=0; i<ROW; i++){
    printf("%-7s",point[i]);
    for (int j=0; j<COLUMN; j++){
        printf("%-7d", road_networks[i][j]);
    } printf("\n");
}

//input
int location, check=0;
while (!check){ //while input for location is not valid
    // could be in one statement but expanded for readability
    printf("\nWhich point are you located?\t0 - A");
    printf("\t1 - B");
    printf("\t2 - C");
    printf("\t3 - D");
    printf("\t4 - E");
    printf("\t5 - F");
    printf("\t6 - G");
    printf("\t7 - H\n");
    printf("Enter your location: ");
    scanf("%d",&location);
    if (location<0 || location>7){ //error for invalid input
        printf("Invalid choice. Enter only numbers 0 to 7.");
    }
    //breaks loop if valid input
    else if (location>=0 || location<=7)
    {
        printf("At point: %s\n",point[location]);
        check=1;
    }
}
for (int i=0; i<COLUMN; i++){
    if (road_networks[location][2]==1){ // directly at C
        printf("Now at point: %s\n",point[2]);
        printf("Arrived at nearest charging station at point: %s",point[2]);
        break;
    }
    else if (road_networks[location][3]==1){ // directly at D
        printf("Now at point: %s\n",point[3]);
        printf("Arrived at nearest charging station at point: %s",point[3]);
        break;
    }
    else if (road_networks[location][i]==1){
        printf("Now at point: %s\n",point[i]);
        if (point[i]==point[2]){
            printf("Arrived at nearest charging station at point: %s",point[i]);
            break;
        }
        else if (point[i]==point[3]){
            printf("Arrived at nearest charging station at point: %s",point[i]);
            break;
        }
        else{
            location=i; //returns if point not found
        }
```

```
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
Enter your location: 7
At point: H
Now at point: F
Now at point: [C]
Arrived at nearest charging station at point: [C]
```