

Github link: <https://github.com/jarenmatthew/CMSC-21/tree/main/Lecture6-7/Assignments>

1a.

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

1b.

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

2.

```
1  #include <string.h>
2  #include <stdbool.h>
3  #include <stdio.h>
4
5  #define NUM_STRINGS ((int) (sizeof(road_network) / sizeof(road_network[0])))
6
7  int main(){
8
9      int station, i = 0, road_network[8][8] = {
10          {1,1,0,0,0,1,0,0},
11          {1,1,1,0,0,0,0,0},
12          {0,1,1,0,1,1,0,0},
13          {0,0,0,1,1,0,0,0},
14          {0,0,0,1,1,0,0,0},
15          {1,0,1,0,0,1,0,0},
16          {1,0,0,1,0,0,1,0},
17          {0,0,0,0,0,1,0,1}
18      };
19
20      char *points[NUM_STRINGS] = {"A", "B", "C", "D", "E", "F", "G", "H"};
21
22      //displays the adjacency matrix
23      printf ("\t A\t B\t [C]\t [D]\t E\t F\t G\t H\n");
24      for (int row = 0; row < NUM_STRINGS; row++){
25          if (points[row] == "C" || points[row] == "D")
26          {
27              printf("[%s]",points[row]);
28          }else{
29              printf("%s ",points[row]);
30          }
31          //prints numbers in matrix
32          for (int col = 0; col < NUM_STRINGS; col++){
33              printf("%8d", road_network[row][col]);
34          }
35          printf("\n");
36      }
37
38      //input for stating location
39      printf("Which point are you located? 0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H\n");
40      scanf ("%d",&station);
41
42
43      if (station < 0 || station > 7){
44          printf("Invalid. Try again.");
45      }
46
47      // if starting location is beside/nearest to charging station
48      else if (road_network[station][2]){
49          printf("At point: %s\n",points[station]);
50          printf("You arrived at charging station %s", points[2]);
51      }else if (road_network[station][3]){
52          printf("At point: %s\n",points[station]);
53          printf("You arrived at charging station %s", points[3]);
54      }
55
56      // for further starting locations
57      else{
58          printf("At point: %s\n",points[station]);
59
60          // Error fix: So the program wont print or stay in location A twice.
61          if (station == 0){
62              i = 1;
63          }
64
65          for (i; i < NUM_STRINGS; i++){
66              if (road_network[station][i])
67              {
68                  printf("Now at point %s \n", points[i]);
69
70                  //if charging station is reached (2 - C, 3 - D)
71                  if (i == 2 || i == 3){
72                      printf("You arrived at charging station %s", points[i]);
73                      break;
74                  }else{
75                      station = i;
76                  }
77              }
78          }
79
80          //goes back to 0 if it reaches the end and no path/point is found
81          else if (i == 7){
82              i = 0;
83          }
84      }
85  }
```

Takeaway: Most of the time I compare C with python since python was the first language I learned. When learning or writing code in C it felt like I had to search more about different things because it seems that you have to include more directives in order to do what you want to do and that C is lengthier. Unlike in python where it feels like plug and play. But I'm not complaining since it is part of the learning process and that you learn more from reading code rather than writing it.