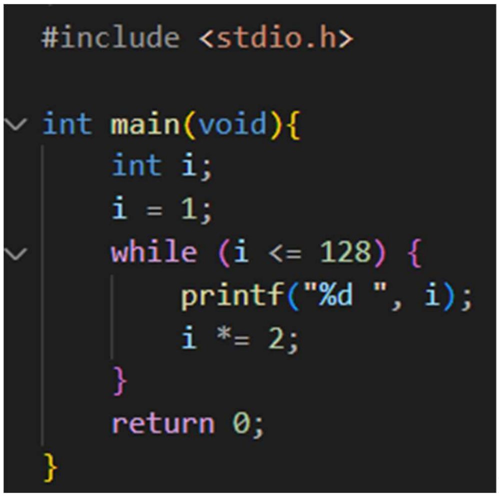
Darah Via D. Moscoso

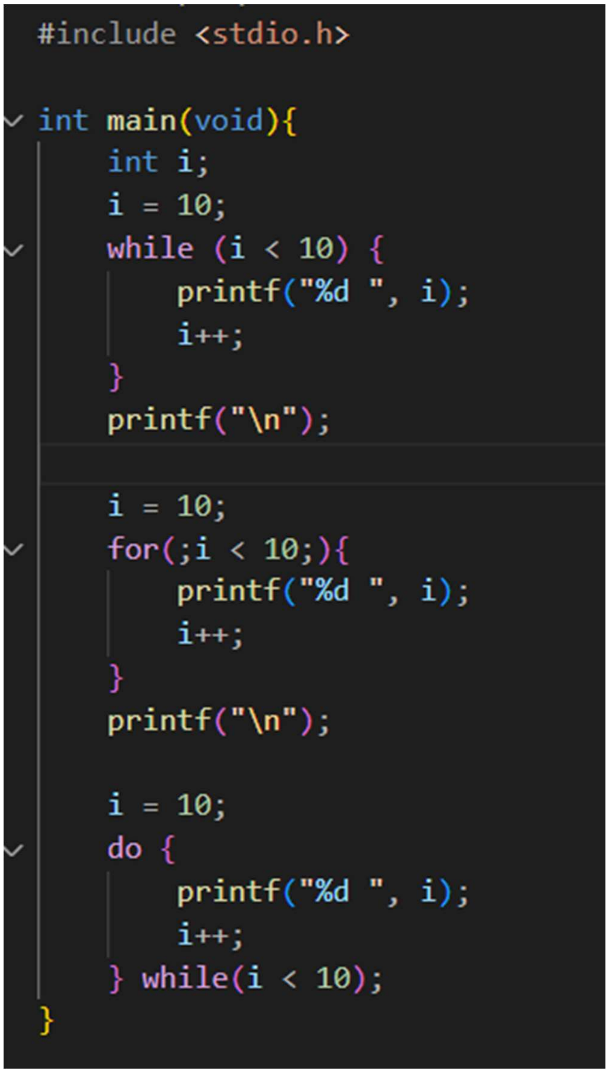
B.S. in Computer Science I

2022 – 08331

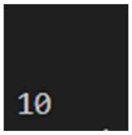
1. 

```c
#include <stdio.h>

int main(void){
    int i;
    i = 1;
    while (i <= 128) {
        printf("%d ", i);
        i *= 2;
    }
    return 0;
}
```

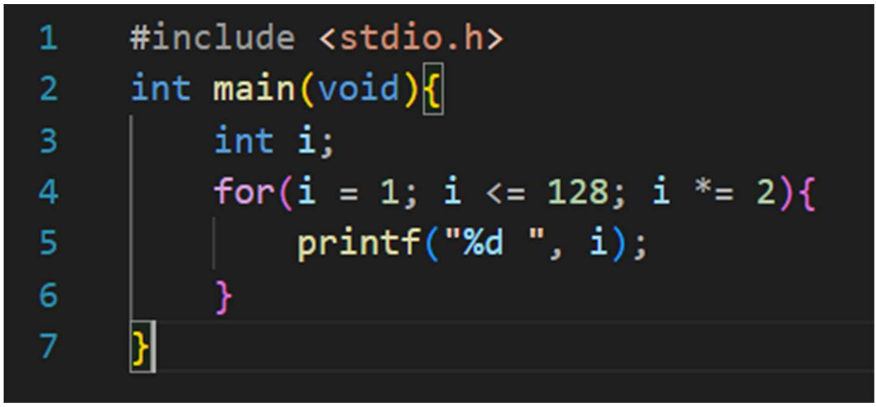
```
1 2 4 8 16 32 64 128
```

2. 

```c
#include <stdio.h>

int main(void){
    int i;
    i = 10;
    while (i < 10) {
        printf("%d ", i);
        i++;
    }
    printf("\n");

    i = 10;
    for(;i < 10;){
        printf("%d ", i);
        i++;
    }
    printf("\n");

    i = 10;
    do {
        printf("%d ", i);
        i++;
    } while(i < 10);
}
```

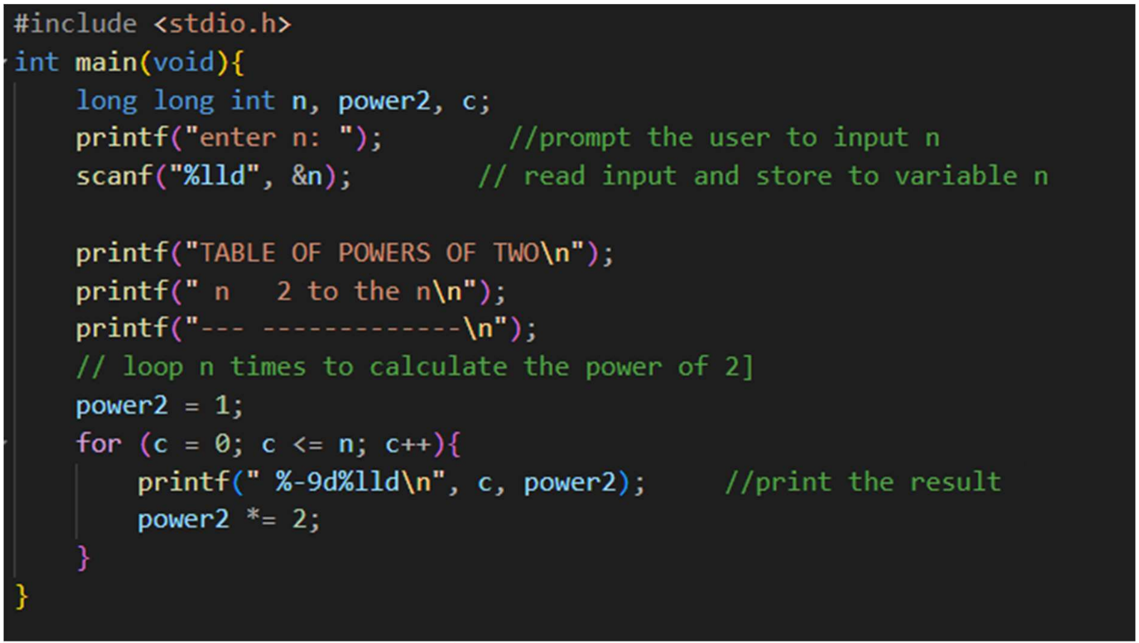
c) do {…} while (i < 10); the loop body of the first two statements will not be executed if the value of i does not satisfy the condition, only the do – while. The output will be 

```
10
```

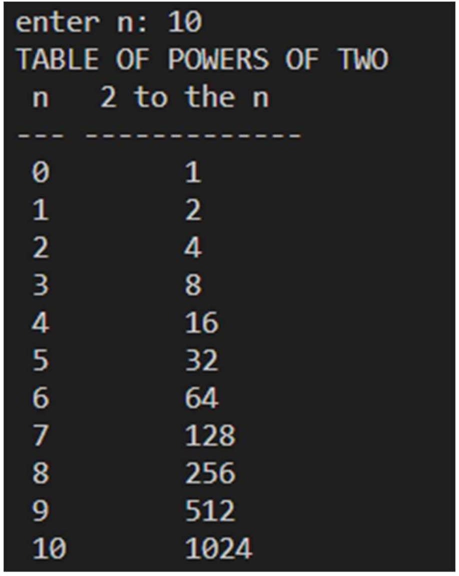
3. 

```c
#include <stdio.h>
int main(void){
    int i;
    for(i = 1; i <= 128; i *= 2){
        printf("%d ", i);
    }
}
```

both of the while and for statements have the same output of "1 2 4 8 16 32 64 128"

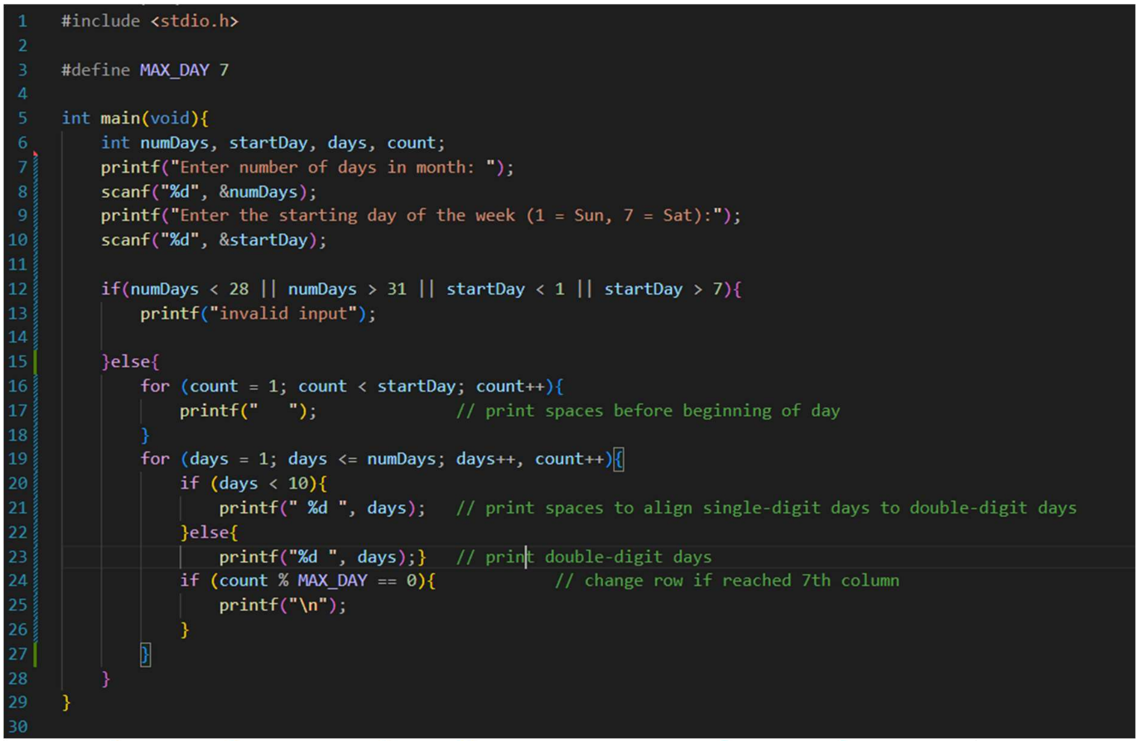
4. 

```c
#include <stdio.h>
int main(void){
    long long int n, power2, c;
    printf("enter n: ");          //prompt the user to input n
    scanf("%lld", &n);          // read input and store to variable n

    printf("TABLE OF POWERS OF TWO\n");
    printf(" n   2 to the n\n");
    printf("--- -------------\n");
    // loop n times to calculate the power of 2]
    power2 = 1;
    for (c = 0; c <= n; c++){
        printf(" %-9d%lld\n", c, power2);     //print the result
        power2 *= 2;
    }
}
```

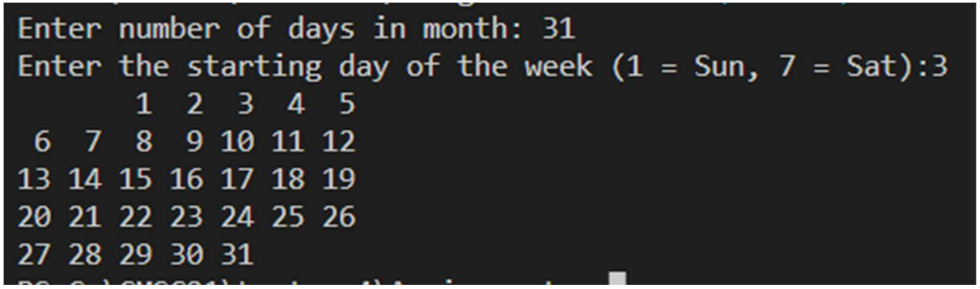


```
enter n: 10
TABLE OF POWERS OF TWO
 n   2 to the n
--- -------------
 0        1
 1        2
 2        4
 3        8
 4        16
 5        32
 6        64
 7        128
 8        256
 9        512
 10       1024
```

5.

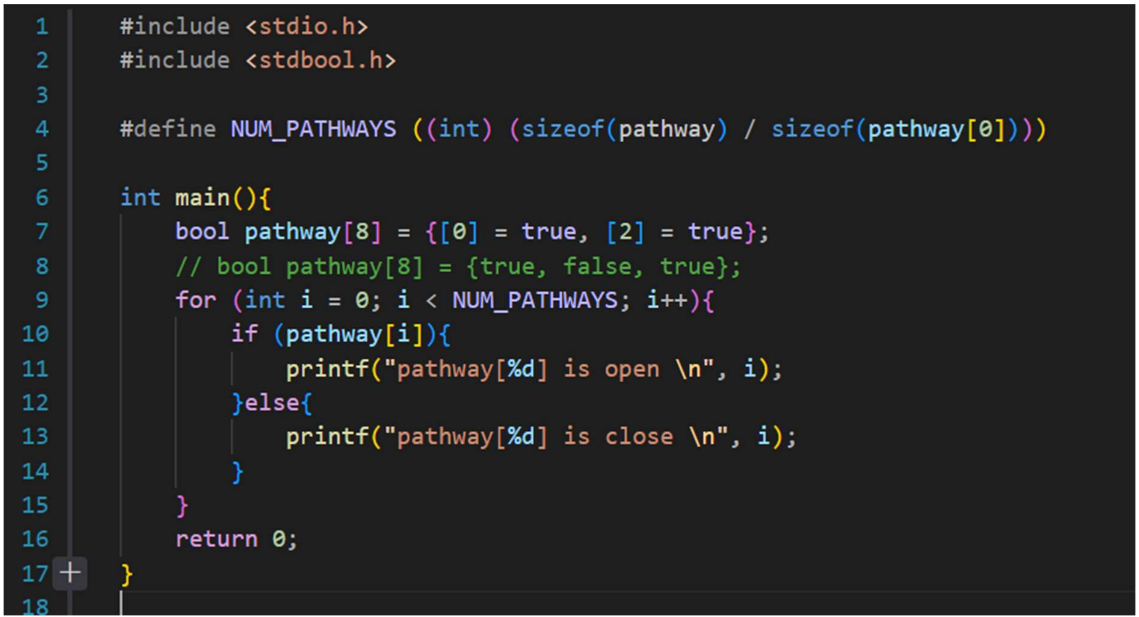


```c
#include <stdio.h>

#define MAX_DAY 7

int main(void){
    int numDays, startDay, days, count;
    printf("Enter number of days in month: ");
    scanf("%d", &numDays);
    printf("Enter the starting day of the week (1 = Sun, 7 = Sat):");
    scanf("%d", &startDay);

    if(numDays < 28 || numDays > 31 || startDay < 1 || startDay > 7){
        printf("invalid input");

    }else{
        for (count = 1; count < startDay; count++){
            printf("   ");              // print spaces before beginning of day
        }
        for (days = 1; days <= numDays; days++, count++){
            if (days < 10){
                printf(" %d ", days);   // print spaces to align single-digit days to double-digit days
            }else{
                printf("%d ", days);}   // print double-digit days
            if (count % MAX_DAY == 0){            // change row if reached 7th column
                printf("\n");
            }
        }
    }
}

```

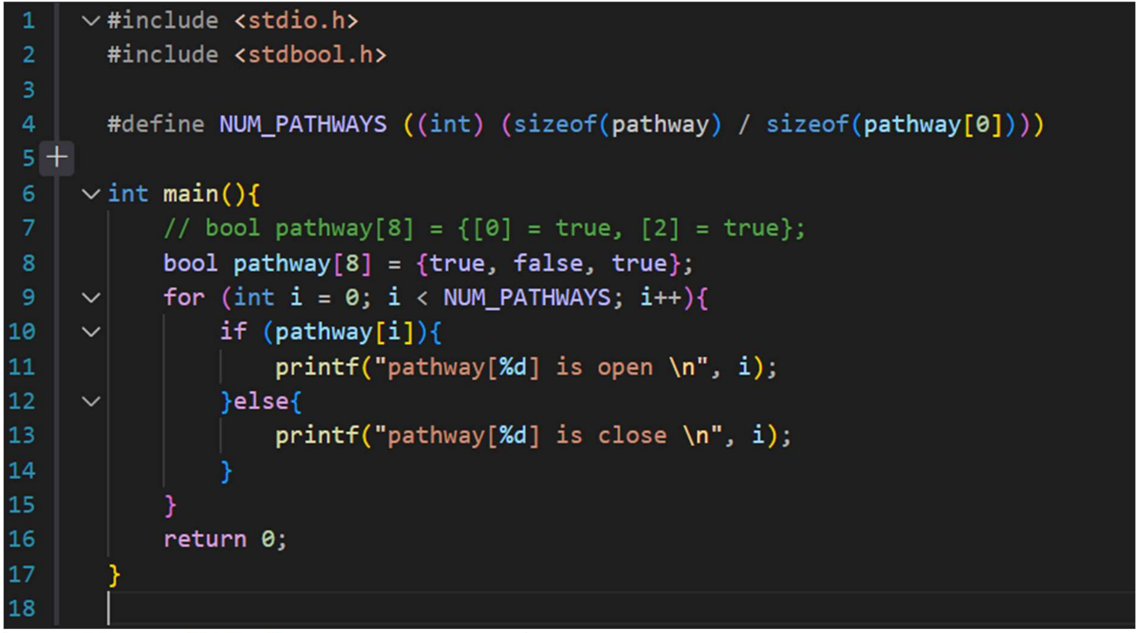


```
Enter number of days in month: 31
Enter the starting day of the week (1 = Sun, 7 = Sat):3
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

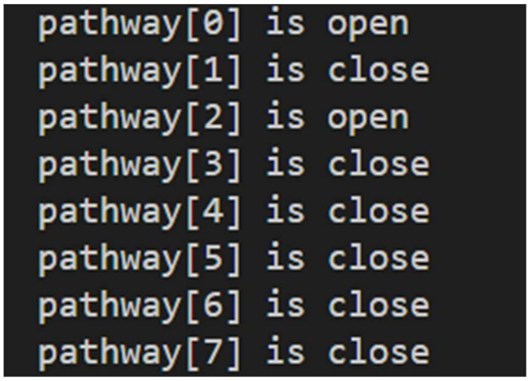
6.



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

#define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))

int main(){
    bool pathway[8] = {[0] = true, [2] = true};
    // bool pathway[8] = {true, false, true};
    for (int i = 0; i < NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathway[%d] is open \n", i);
        }else{
            printf("pathway[%d] is close \n", i);
        }
    }
    return 0;
}

```

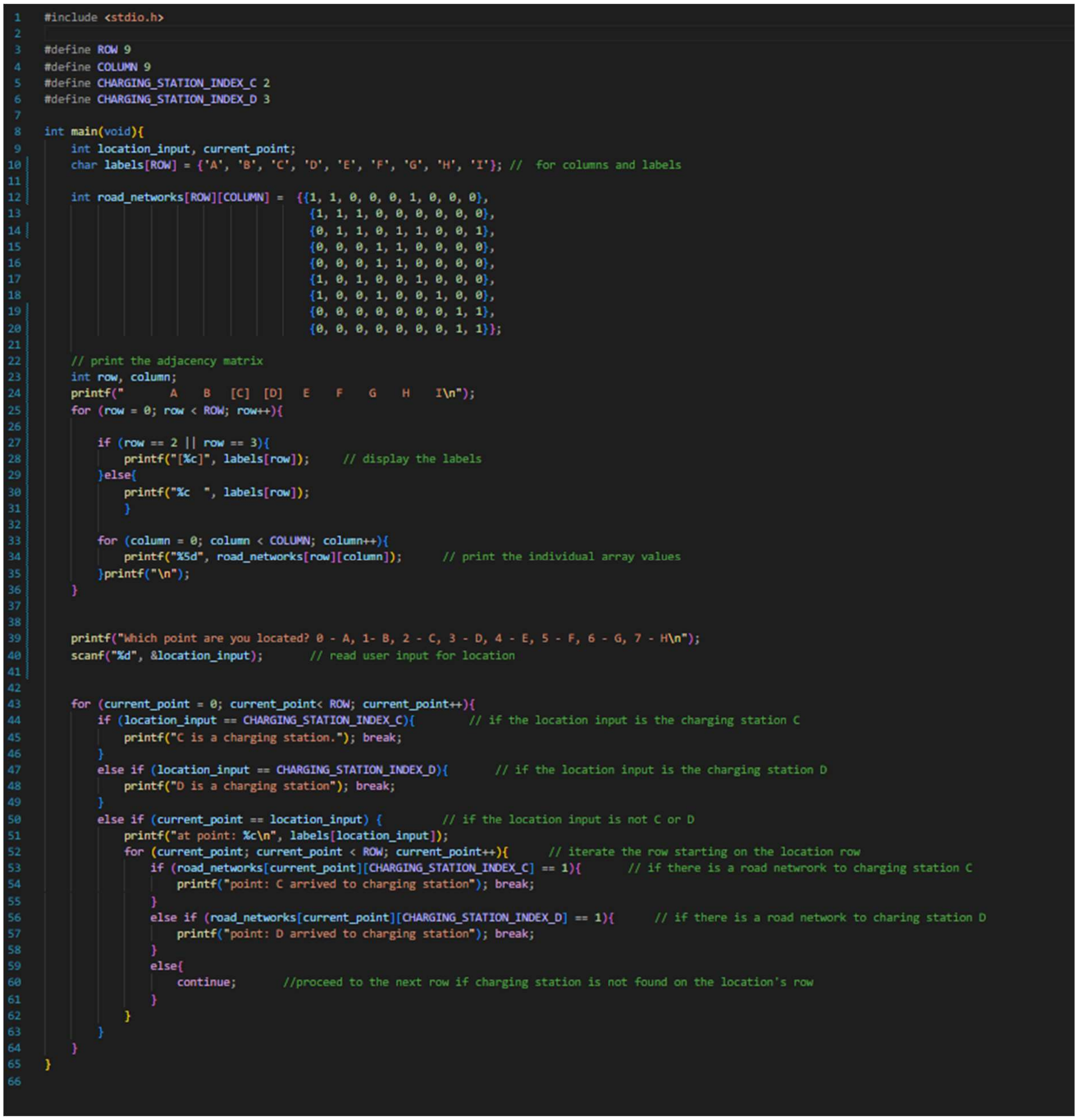


```c
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))

int main(){
    // bool pathway[8] = {[0] = true, [2] = true};
    bool pathway[8] = {true, false, true};
    for (int i = 0; i < NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathway[%d] is open \n", i);
        }else{
            printf("pathway[%d] is close \n", i);
        }
    }
    return 0;
}

```



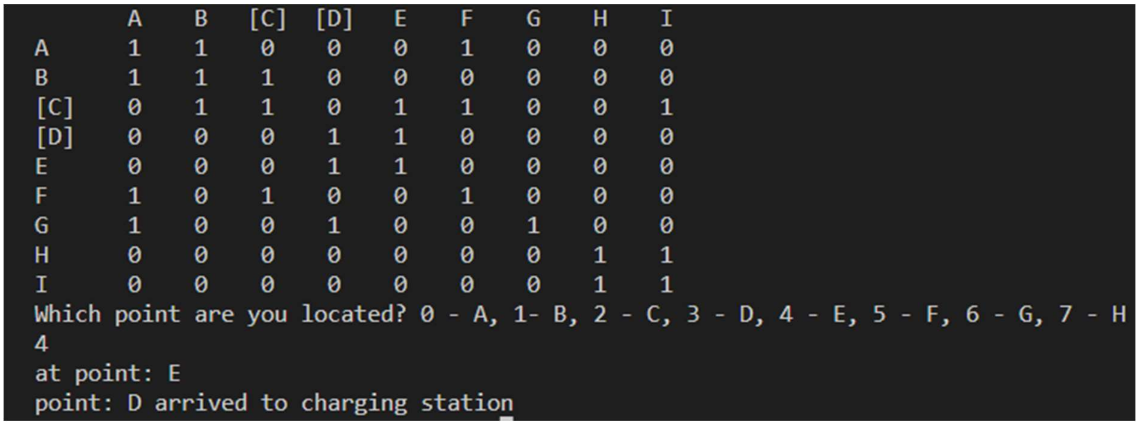
```
pathway[0] is open
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

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



```c
#include <stdio.h>

#define ROW 9
#define COLUMN 9
#define CHARGING_STATION_INDEX_C 2
#define CHARGING_STATION_INDEX_D 3

int main(void){
    int location_input, current_point;
    char labels[ROW] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'}; //  for columns and labels

    int road_networks[ROW][COLUMN] =  {{1, 1, 0, 0, 0, 1, 0, 0, 0},
                                        {1, 1, 1, 0, 0, 0, 0, 0, 0},
                                        {0, 1, 1, 0, 1, 1, 0, 0, 1},
                                        {0, 0, 0, 1, 1, 0, 0, 0, 0},
                                        {0, 0, 0, 1, 1, 0, 0, 0, 0},
                                        {1, 0, 1, 0, 0, 1, 0, 0, 0},
                                        {1, 0, 0, 1, 0, 0, 1, 0, 0},
                                        {0, 0, 0, 0, 0, 0, 0, 1, 1},
                                        {0, 0, 0, 0, 0, 0, 0, 1, 1}};

    // print the adjacency matrix
    int row, column;
    printf("       A    B   [C]  [D]   E    F    G    H    I\n");
    for (row = 0; row < ROW; row++){

        if (row == 2 || row == 3){
            printf("[%c]", labels[row]);     // display the labels
        }else{
            printf("%c  ", labels[row]);
            }

        for (column = 0; column < COLUMN; column++){
            printf("%5d", road_networks[row][column]);      // print the individual array values
        }printf("\n");
    }


    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_input);       // read user input for location


    for (current_point = 0; current_point< ROW; current_point++){
        if (location_input == CHARGING_STATION_INDEX_C){        // if the location input is the charging station C
            printf("C is a charging station."); break;
        }
        else if (location_input == CHARGING_STATION_INDEX_D){       // if the location input is the charging station D
            printf("D is a charging station"); break;
        }
        else if (current_point == location_input) {         // if the location input is not C or D
            printf("at point: %c\n", labels[location_input]);
            for (current_point; current_point < ROW; current_point++){      // iterate the row starting on the location row
                if (road_networks[current_point][CHARGING_STATION_INDEX_C] == 1){       // if there is a road netwrork to charging station C
                    printf("point: C arrived to charging station"); break;
                }
                else if (road_networks[current_point][CHARGING_STATION_INDEX_D] == 1){      // if there is a road network to charing station D
                    printf("point: D arrived to charging station"); break;
                }
                else{
                    continue;       //proceed to the next row if charging station is not found on the location's row
                }
            }
        }
    }
}
```

7.



```
       A    B   [C]  [D]   E    F    G    H    I
A      1    1    0    0    0    1    0    0    0
B      1    1    1    0    0    0    0    0    0
[C]    0    1    1    0    1    1    0    0    1
[D]    0    0    0    1    1    0    0    0    0
E      0    0    0    1    1    0    0    0    0
F      1    0    1    0    0    1    0    0    0
G      1    0    0    1    0    0    1    0    0
H      0    0    0    0    0    0    0    1    1
I      0    0    0    0    0    0    0    1    1
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
point: D arrived to charging station
```