

CMSC 21  
2nd Semester AY 2022-2023  
LECTURE 4  
Jhoanna R. Olana - 202211140

Item 1:

What is the output of the following program?

```
#include <stdio.h>

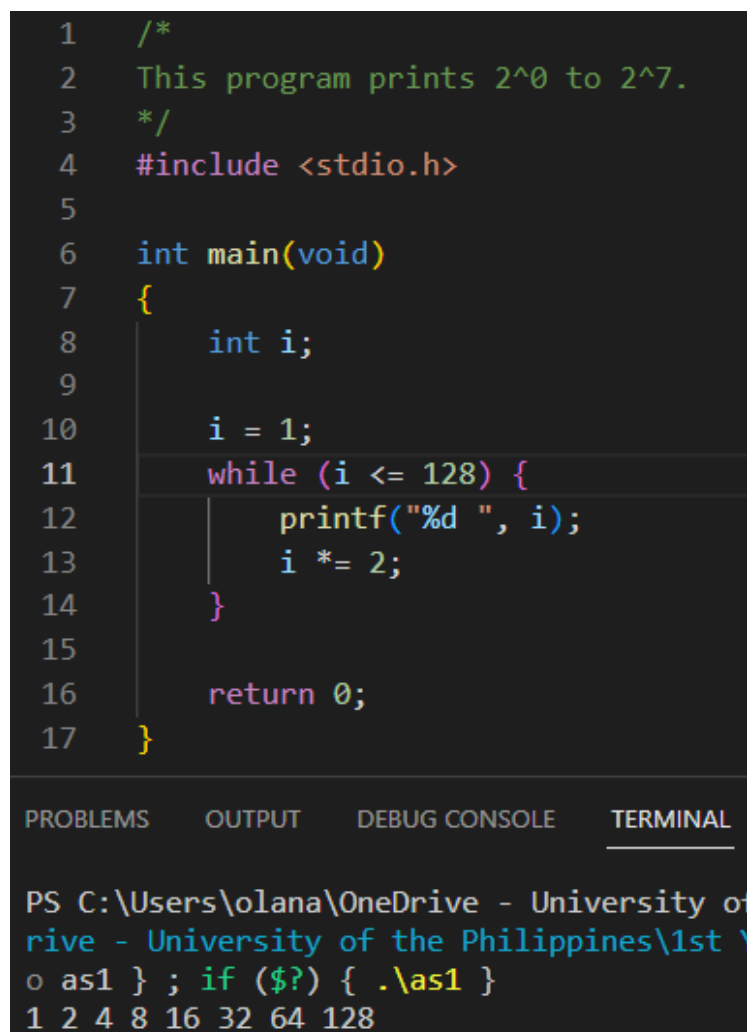
int main(void)
{
    int i;

    i = 1;
    while (i <= 128) {
        printf("%d ", i);
        i *= 2;
    }

    return 0;
}
```

Answer:

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



The image shows a screenshot of a code editor with a dark background. The code is a C program that prints powers of 2 from 2^0 to 2^7. The code is as follows:

```
1  /*
2  This program prints 2^0 to 2^7.
3  */
4  #include <stdio.h>
5
6  int main(void)
7  {
8      int i;
9
10     i = 1;
11     while (i <= 128) {
12         printf("%d ", i);
13         i *= 2;
14     }
15
16     return 0;
17 }
```

Below the code editor, there is a terminal window with the following output:

```
PS C:\Users\olana\OneDrive - University of the Philippines\1st Year\CMSC 21\
o as1 } ; if ($?) { .\as1 }
1 2 4 8 16 32 64 128
```

## Item 2:

Which one of the following statements is not equivalent to the other two (assuming that the loop bodies are the same)?

- a) `while (i < 10) {...}`
- b) `for (; i < 10;) {...}`
- c) `do {...} while (i < 10);`

Given that the loop bodies are the same (including the initialize, condition and update parts), **all are equivalent to each other.**

```
1  /*
2  This program shows how a loop statement may be constructed using several loop controls as
3  long as the portions such as initialize, condition, and update are the same.
4  Written in 2023 by Jhoanna Olana.
5  */
6  #include <stdio.h>
7
8  int main(void) {
9      int i = 0;                // Initialize
10
11      // ---- while loop ----
12      // while (i < 10) {        // Condition
13      //     printf("%d ", i);
14      //     i++;                // Update
15      // }
16
17      // ---- for loop ----
18      // for ( ; i < 10; ) {      // Condition
19      //     printf("%d ", i);
20      //     i++;                // Update
21      // }
22
23      // ---- do-while loop ----
24      // do {
25      //     printf("%d ", i);
26      //     i++;                // Update
27      // } while (i < 10);        // Condition
28
29      return 0;
30 }
```

```

8  int main(void) {
9      int i = 0;
10
11     // ---- while loop ----
12     // while (i < 10) {
13     //     printf("%d ", i);
14     //     i++;
15     // }
16
17     // ---- for loop ----
18     // for ( ; i < 10; ) {
19     //     printf("%d ", i);
20     //     i++;
21     // }
22
23     // ---- do-while loop ----
24     do {
25         printf("%d ", i);
26         i++;
27     } while (i < 10);
28
29     return 0;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assign  
0 1 2 3 4 5 6 7 8 9

```

8  int main(void) {
9      int i = 0;
10
11     // ---- while loop ----
12     // while (i < 10) {
13     //     printf("%d ", i);
14     //     i++;
15     // }
16
17     // ---- for loop ----
18     for ( ; i < 10; ) {
19         printf("%d ", i);
20         i++;
21     }
22
23     // ---- do loop ----

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st  
o as2 } ; if (\$?) { .\as2 }  
0 1 2 3 4 5 6 7 8 9

```

8  int main(void) {
9      int i = 0;
10
11     // ---- while loop ----
12     while (i < 10) {
13         printf("%d ", i);
14         i++;
15     }
16
17     // ---- for loop ----

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assign  
) { .\as2 }  
0 1 2 3 4 5 6 7 8 9

Item 3:

Convert item 1 into an equivalent for statement. You can validate your answer by checking if the produced outputs by both the while and for statements are similar.

```
1  /*
2  This program prints 2^0 to 2^7 using a for loop.
3  Written in 2023 by Jhoanna Olana.
4  */
5  #include <stdio.h>
6
7  int main(void) {
8      int i;
9
10     /*
11     Prints numbers 1(2^0) to 128(2^7). In every iteration, the value of i is updated by multiplying its old value
12     by 2. Unlike in item 1, the initialize and update portions of the loop are all in the same line.
13     */
14
15     for (i = 1; i <= 128; i *= 2) {
16         printf("%d ", i);
17     }
18
19     return 0;
20 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21> cd "c:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if (\$?) { gcc as3.c -o as3 } ; if (\$?) { .\as3 }

1 2 4 8 16 32 64 128

#### Item 4:

Write a code that computes for the power of two:

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

```

1  /*
2  This program computes the power of two from 0 to n.
3  Written in 2023 by Johanna Olana.
4  */
5  #include <stdio.h>
6  #include <math.h>
7
8  int main(void) {
9      // ----- Initialization -----
10
11      /*
12      n(the last number for which 2 will be raised) and i(the iterator variable) are all declared as integers
13      while pow2(stores the powers of 2) is declared as double so that it can handle large numbers (such as > 2^30).
14      */
15
16      int n, i;
17      double pow2;
18
19      // ----- Reads inputs and displays headers -----
20      printf("TABLE OF POWERS OF TWO\n");
21
22      printf("Enter a number (n): ");
23      scanf("%d", &n);
24
25      printf("n\t2 to the n\n");
26      printf("---\t-----\n");
27
28      // ----- Prints the powers of 2 from 0 to n -----
29
30      /*
31      Uses the pow function to print each power.
32      Since pow2 is a double, "%.1f" or "%.01f" ensures that no decimal places are printed for each power.
33      */
34
35      for (i = 0; i <= n; i++) {
36          pow2 = pow(2, i);
37          printf("%d\t%.1f\n", i, pow2);
38      }
39
40      return 0;
41 }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21> cd "c:\Users\olana\OneDrive
Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if ($?) { gcc as4.c -o as4 } ; if ($?) { .\as4 }

```

TABLE OF POWERS OF TWO

Enter a number (n): 10

```

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

```

## Item 5:

Write a program that displays a one-month calendar.

```
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
```

There should be a user prompt to set:

- The number of days.
- The day of the week on which the month begins.

Additionally, add checkers to validate whether the days entered are valid. For instance, the following number of days are invalid: 32, -1, 0, 27.

```
1  /*
2  This program displays a one-month calendar based on the user's selection of the number of days and the day
3  of the week when the month begins.
4  Written in 2023 by Jhoanna Olana.
5  */
6  #include <stdio.h>
7
8  int main(void) {
9      // ----- Initialization -----
10
11      /*
12      A month calendar can be viewed as a table, with a maximum dimension of 5x7. Furthermore, its empty cells
13      can be considered containing "0" day. Since we only want a placeholder for the empty cells and cells that
14      contain month days, we use an 1d array with a size of 35.
15      The use of the other variables will be explained later.
16      */
17
18      int monthDay, startDay, count, i, j;
19      int month[35] = {0};
20
21      // ----- Reads inputs -----
22
23      /*
24      Do while validates the data that users entered.
25      */
26
27      do {
28          printf("Enter number of days in a month: ");
29          scanf("%d", &monthDay);
30      } while (monthDay < 28 || monthDay > 31);
31
```

```

32     do {
33         printf("Enter the starting day of the week (1 = Sun, 7 = Sat): ");
34         scanf("%d", &startDay);
35     } while (startDay < 1 || startDay > 7);
36
37
38     // ----- Generates the one-month calendar -----
39
40     /*
41     Starting from the (startDay - 1) index (j), every element in the array will be replaced with 1 to monthdays (i).
42     */
43
44     j = startDay - 1;
45     for (i = 1; i <= monthDay; i++) {
46         month[j] = i;
47         j++;
48     }
49

```

```

50     // ----- Prints the one-month calendar -----
51
52     printf("\n%c%c%c%c%c%c%c\n", 'S', 'M', 'T', 'W', 'T', 'F', 'S');
53
54     /*
55     To have a uniform look, every element in the array (except 0) will be printed with a minimum size of 3 characters (%3d)
56     -- 0 is replaced with 3 spaces. Count variable ensures that there are only 7 days printed in every row/week (including "0"
57     */
58
59     count = 0;
60     for (i = 0; i < 35; i++) {
61
62         if (month[i] == 0) {
63             printf("   ");
64         } else {
65             printf("%3d", month[i]);
66         }
67         count++;
68
69         if (count == 7) {
70             printf("\n");
71             count = 0;
72         }
73
74     }
75
76     return 0;
77

```

```
50 // ----- Prints the one-month calendar -----
51
52 printf("\n%c%c%c%c%c%c%c\n", 'S', 'M', 'T', 'W', 'T', 'F', 'S');
53
54 /*
55 To have a uniform look, every element in the array (except 0) will be printed with a minimum size of
56 -- 0 is replaced with 3 spaces. Count variable ensures that there are only 7 days printed in every row
57 */
58
59 count = 0;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21> cd "c:\Users\olana\OneDrive Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if (\$?) { gcc as5.c -o as5 } ; if (\$?) { .\as5 }

Enter number of days in a month: 31

Enter the starting day of the week (1 = Sun, 7 = Sat): 3

S	M	T	W	T	F	S
		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		

## Item 6:

In the program below, an array named pathway contains eight bool values. Each bool element refers to whether a pathway is open or closed for transportation. Only pathways 0 and 2 are open while the rest are still close due to road constructions and fixings.

```
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] = {true, false, true, false, false, false, false, false};
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 }
```



- a. Revise line 16 such that you use a designated initializer to set pathways 0 and 2 to true, and the rest will be false. Make the initializer as short as possible.
- b. Revise line 16 such that the initializer will be short as possible (without using a designated initializer)

```
1  /*
2  This program demonstrates the different ways we can initialize an array.
3  6a-6b are written by Jhoanna Olana.
4  */
5  #include <stdio.h>
6  #include <stdbool.h>
7
8  #define NUM_PATHWAYS ( (int) (sizeof(pathway)) / (sizeof(pathway[0])) )
9
10 int main(void) {
11
12     // ----- Original -----
13     bool pathway[8] = {true, false, true, false, false, false, false};
14
15     // ----- 6.a -----
16     /*
17     Uses designated initializers. The rest will automatically be given a value of false.
18     */
19
20     // bool pathway[8] = {[0] = true, [2] = true};
21
22     // ----- 6.b -----
23     /*
24     Since the array consists of true values only on its first three elements, we can initialize it by providing
25     the values of those elements. The rest will be automatically assigned the value false.
26     */
27
28     // bool pathway[8] = {true, false, true, };
29
30     // ----- Displays the status of each pathway/elements in the array -----
31
32     for (int i = 0; i < NUM_PATHWAYS; i++) {
33         if (pathway[i]) {
34             printf("pathway[%d] is open \n", i);
35         } else {
36             printf("pathway[%d] is close \n", i);
37         }
38     }
39
40     return 0;
41 }
```

```

12 // ----- Original -----
13 bool pathway[8] = {true, false, true, false, false, false, false};
14

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if ($?) { gcc as6.c -o as6 } ;
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


```

```

15 // ----- 6.a -----
16 /*
17 Uses designated initializers. The rest will automatically be given a value of false.
18 */
19
20 bool pathway[8] = {[0] = true, [2] = true};
21

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

 Code

```

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21> cd
rive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; i
o as6 } ; if ($?) { .\as6 }
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

```

```

22 // ----- 6.b -----
23 /*
24 Since the array consists of true values only on its first three elements, we can initialize it by providing
25 the values of those elements. The rest will be automatically assigned the value false.
26 */
27
28 bool pathway[8] = {true, false, true, };
29

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL



```

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21> cd "c:\Users\olana\OneDrive - Univers
Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if ($?) { gcc as6.c -o as6 } ; if ($?) { .\as6 }
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

```

## Item 7:

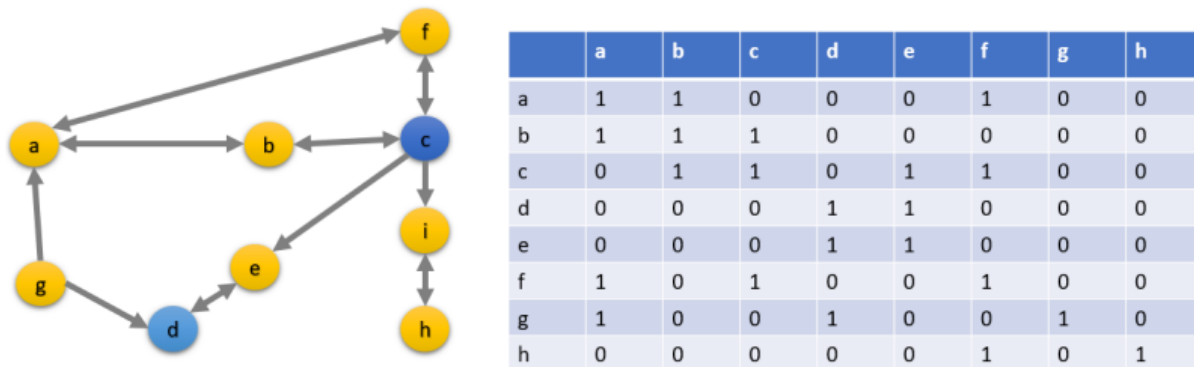
A road network can be represented using graphs. Assuming we have points / stations a, b, c, d, e, f, g, and h, we can represent a direct path from a point to another point using arrows.

For example, based on the graph below:

- There is a two-way path between point a and point b, point a and point f, point f and point c, and point d and e.
- There is a one-way path from point c to point i but no direct path between point i to point c.

All of the nodes are points/destinations, but the blue ones specifically represent charging stations. The road network between these points/destinations can be represented using an adjacency matrix of Booleans (0s and 1s), as shown below.

For instance,  $a \rightarrow b = 1$  and  $b \rightarrow a = 1$  given that there's a two-way direct path between a and b. Meanwhile,  $a \rightarrow c = 0$  since there is no direct path between a and c. Moreover,  $a \rightarrow g = 0$  but  $g \rightarrow a = 1$  since there is a one-way path from point g to point a.



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.

```

1  /*
2  This program performs the 7th task listed in this document:
3  https://upsystem-my.sharepoint.com/:b:/g/personal/jrolana\_outlook\_up\_edu\_ph/Ecnoqciqh11Bo40WHZvzFABLm04AA9Mvg0qdf\_JcYpHlw
4  Written in 2023 by Jhoanna Olana.
5  */
6  #include <stdio.h>
7
8  #define NUM_POINTS 9
9
10 int main(void) {
11     // ----- Initialization -----
12     /*
13     Adjacency matrix of the given graph (node/point i included).
14     */
15     int road_networks[NUM_POINTS][NUM_POINTS] = {
16         {1, 1, 0, 0, 0, 1, 0, 0, 0},
17         {1, 1, 1, 0, 0, 0, 0, 0, 0},
18         {0, 1, 1, 0, 1, 1, 0, 0, 0},
19         {0, 0, 0, 1, 1, 0, 0, 0, 0},
20         {0, 0, 0, 1, 1, 0, 0, 0, 0},
21         {1, 0, 1, 0, 0, 1, 0, 0, 0},
22         {1, 0, 0, 1, 0, 0, 1, 0, 0},
23         {0, 0, 0, 0, 0, 0, 0, 1, 1},
24         {0, 0, 0, 0, 0, 0, 0, 1, 1},
25     };
26
27     char points[NUM_POINTS] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i'};
28
29     int r, c, point;
30

```

```

31     // ----- Displays the matrix -----
32
33     // Column Header
34     /*
35     Every element in the array named points will be printed as the column header of the adjacency matrix.
36     Charging stations c and d are enclosed in brackets.
37     */
38     printf("\t");
39     for (r = 0; r < NUM_POINTS; r++) {
40         if (points[r] == 'c' || points[r] == 'd') {
41             printf("[%c]\t", points[r]);
42         } else {
43             printf("%c\t", points[r]);
44         }
45     }
46     printf("\n");
47

```

```

48 // Row
49 /*
50 Every row in the adjacency matrix consists of the points (act as the headers for the rows)
51 and the edges for each point (or the elements in every row of the array named road_networks).
52 */
53 for (r = 0; r < NUM_POINTS; r++) {
54     // Row Header
55     if (points[r] == 'c' || points[r] == 'd') {
56         printf("[%c]\t", points[r]);
57     } else {
58         printf("%c\t", points[r]);
59     }
60
61     // Elements in each row
62     for (c = 0; c < NUM_POINTS; c++) {
63         printf("%d\t", road_networks[r][c]);
64     }
65
66     printf("\n");
67 }
68 printf("\n");
69

```

```

70 // ----- Determines the nearest charging station given a point. -----
71 /*
72 Do while ensures that users enter a known point only. Nested if-else statements give the
73 appropriate charging station for each point based on how it was constructed in the graph.
74 */
75 do {
76     printf("Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8- I): ");
77     scanf("%d", &point);
78 } while (point < 0 || point > 8);
79
80 printf("At point: ");
81
82 if (point == 0 || point == 1 || point == 5) {
83
84     if (point == 0) {
85         printf("A\n");
86     } else if (point == 1) {
87         printf("B\n");
88     } else {
89         printf("F\n");
90     }
91
92     printf("point: C arrived to charging station.\n");
93 }

```

```

94     else if (point == 2) {
95         printf("C\npoint: C is a charging station.\n");
96     }
97     else if (point == 3) {
98         printf("D\npoint: D is a charging station.\n");
99     }
100    else if (point == 6 || point == 4) {
101
102        if (point == 6) {
103            printf("G\n");
104        } else {
105            printf("E\n");
106        }
107
108        printf("point: D arrived to charging station.\n");
109    }
110    else {
111
112        if (point == 7) {
113            printf("H\n");
114        } else {
115            printf("I\n");
116        }
117
118        printf("point: No charging station found.\n");
119    }
120
121    return 0;
122 }

```

```

15    int road_networks[NUM_POINTS][NUM_POINTS] = {
16        {1, 1, 0, 0, 0, 1, 0, 0, 0},
17        {1, 1, 1, 0, 0, 0, 0, 0, 0},
18        {0, 1, 1, 0, 1, 1, 0, 0, 0},

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21> cd "c:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if (\$?) { gcc as7.c -o as7 } ; if (\$?) { .\as7 }

	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	0
[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, 8- I): 0

At point: A

point: C arrived to charging station.