

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>

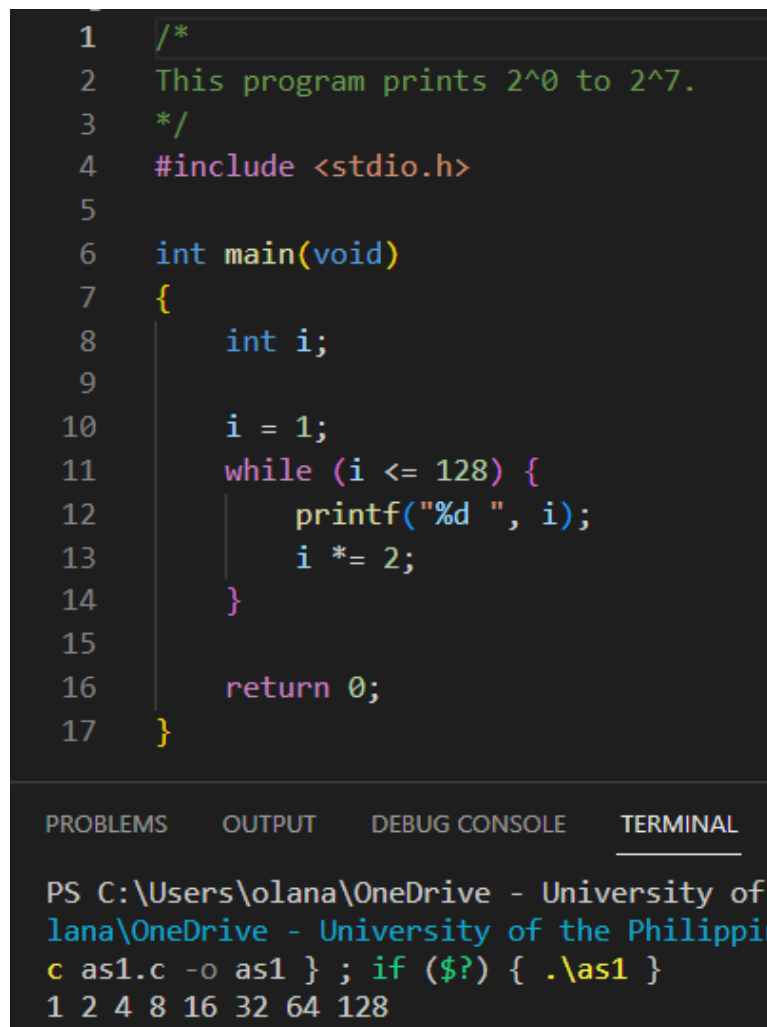
int main(void)
{
    int i;

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

    return 0;
}
```

Answer:

1 2 4 8 16 32 64 128



The screenshot shows a code editor with a C program and a terminal window below it. 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 }
```

The terminal window shows the command prompt and the output of the program:

```
PS C:\Users\olana\OneDrive - University of the Philippines - Manila> gcc as1.c -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 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;                // Initialize
10
11     // ---- while loop ----
12     while (i < 10) {           // Condition
13         printf("%d ", i);
14         i++;                   // Update
15     }

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

21\Lecture4> cd "c:\Users\olana\OneDrive - University of the Ph
m\CMSC 21\CMSC21\Lecture4\Assignments\" ; if ($?) { gcc as2.c -

```

0 1 2 3 4 5 6 7 8 9

```

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 ----

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\olana\OneDrive - University of the Philippines\1st
lana\OneDrive - University of the Philippines\1st Year - 2nd Se
c as2.c -o as2 } ; if ($?) { .\as2 }

```

0 1 2 3 4 5 6 7 8 9

```
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;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year
ines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments\" ; if C
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      /*
12      Prints numbers 1(2^0) to 128(2^7). In every iteration, the value of i is updated by multiplying its old value
13      by 2. Unlike in item 1, the initialize and update portions of the loop are all in the same line.
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\Lecture4> 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 powers of two from 0 to n.
3  Written in 2023 by Jhoanna Olana.
4  */
5  #include <stdio.h>
6  #include <math.h>
7
8  int main(void) {
9      // ----- Initialization -----
10     /*
11      n(the last number for which 2 will be raised) and i(the iterator variable) are all declared as integers,
12      while pow2(stores the powers of 2) is declared as double so that it can handle large numbers (such as  $2^{30}$ ).
13      */
14
15     int n, i;
16     double pow2;
17
18     // ----- Reads inputs and displays headers -----
19     printf("TABLE OF POWERS OF TWO\n");
20
21     printf("\nEnter a number (n): ");
22     scanf("%d", &n);
23
24     printf("\n\n\t2 to the n\n");
25     printf("---\t-----\n");
26

```

```

27     // ----- Prints the powers of 2 from 0 to n -----
28     /*
29     Pow2 is first set to be 1, and in subsequent iterations, it is set to the result of multiplying its old value by 2.
30     Since pow2 is a double, "%.1f" or "%.01f" ensures that no decimal places are printed for each power.
31     */
32
33     for (i = 0; i <= n; i++) {
34         if (i == 0) {
35             pow2 = 1;
36         } else {
37             pow2 *= 2;
38         }
39         printf("%d\t%.1f\n", i, pow2);
40     }
41
42     return 0;
43 }

```

```
19     printf("TABLE OF POWERS OF TWO\n");
20
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\olana\OneDrive - University of the Phil:
ines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Ass:
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  #define SIZE 42
9
10 int main(void) {
11     // ----- Initialization -----
12     /*
13     A one-month calendar can be viewed as a table, with a maximum dimension of 6x7, thus we can use an array to
14     create it. Since we only want a placeholder for the empty cells(containing "0" day) and cells that contain
15     month days, we use an 1d array with a size of 42.
16     */
17
18     int monthDay, startDay, count, i, j;
19     int month[SIZE] = {0};
20
21     // ----- Reads inputs -----
22     /*
23     Do while validates the data that users entered.
24     */
25
26     do {
27         printf("Enter number of days in a month: ");
28         scanf("%d", &monthDay);
29     } while (monthDay < 28 || monthDay > 31);
30

```

```

31     do {
32         printf("Enter the starting day of the week (1 = Sun, 7 = Sat): ");
33         scanf("%d", &startDay);
34     } while (startDay < 1 || startDay > 7);
35
36
37     // ----- Generates the one-month calendar -----
38     /*
39     Starting from the (startDay - 1) index (j), every element in the array will be replaced with 1 to monthdays (i).
40     */
41
42     j = startDay - 1;
43     for (i = 1; i <= monthDay; i++) {
44         month[j] = i;
45         j++;
46     }
47
48     // ----- Prints the one-month calendar -----
49
50     printf("\n%c%c%c%c%c%c%c%c\n", 'S', 'M', 'T', 'W', 'T', 'F', 'S');
51

```



```

52  /*
53  To have a uniform look, every element in the array (except 0) will be printed with a minimum size of 3 characters (%3d)
54  -- 0 is replaced with 3 spaces. Count variable ensures that there are only 7 days printed in every row/week (including "0")
55  */
56
57  count = 0;
58  for (i = 0; i < SIZE; i++) {
59
60      if (month[i] == 0) {
61          printf("   ");
62      } else {
63          printf("%3d", month[i]);
64      }
65      count++;
66
67      if (count == 7) {
68          printf("\n");
69          count = 0;
70      }
71  }
72  }
73
74  return 0;
75

```

```

48  // ----- Prints the one-month calendar -----
49

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Year\olana\OneDrive - University of the Philippines\1st Year - 2nd Year\c 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.
18     Since the array is of boolean type, its elements can be written as 0(false) or 1(true).
19     */
20
21     bool pathway[8] = {[0] = 1, [2] = 1};
22
23     // ----- 6.b -----
24     /*
25     Because it has the value 0 from the fourth to the last element, we can initialize it by providing the
26     first three elements only. The rest will be automatically assigned the value false.
27     */
28
29     // bool pathway[8] = {1, 0, 1};
30

```

```

31     // ----- Displays the status of each pathway/elements in the array -----
32
33     for (int i = 0; i < NUM_PATHWAYS; i++) {
34         if (pathway[i]) {
35             printf("pathway[%d] is open \n", i);
36         } else {
37             printf("pathway[%d] is close \n", i);
38         }
39     }
40
41     return 0;
42 }

```

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4\Assignments> cd "c:\Users\olana\OneDrive - University of the Philippines\1st 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
```

```
15 // ----- 6.a -----
16 /*
17 Uses designated initializers.
18 Since the array is of boolean type, its elements can be written as 0(false) or 1(true).
19 */
20
21 bool pathway[8] = {[0] = 1, [2] = 1};
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\olana\OneDrive - University of the Philippines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lecture4> cd "c:\Users\olana\OneDrive - University of the Philippines\1st 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
```

```
23 // ----- 6.b -----
24 /*
25  Because it has the value 0 from
26  first three elements only. The
27  */
28
29 bool pathway[8] = {1, 0, 1};
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\olana\OneDrive - University of t
ines\1st Year - 2nd Sem\CMSC 21\CMSC21\Lectu
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:

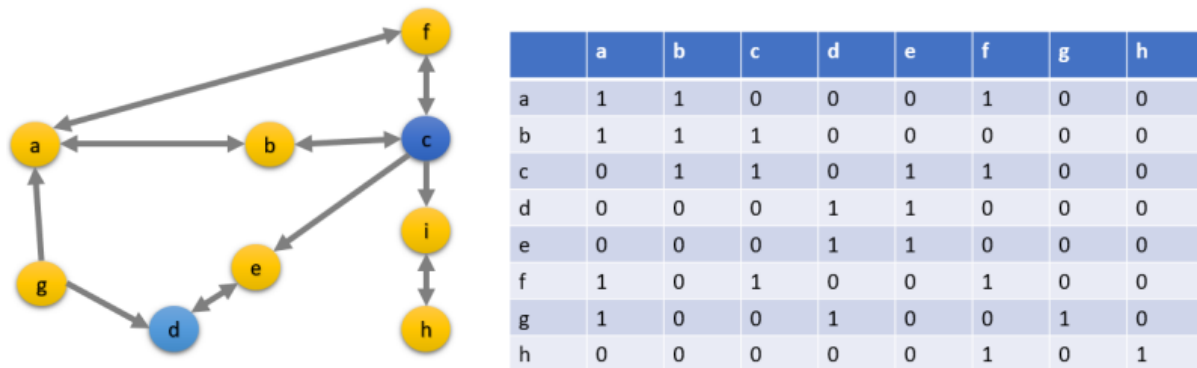
- There is a two-way path between point a and point b, point a and point f, point f and point c, and point d and e.
- There is a one-way path from point c to point i but no direct path between point i to point c.

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/E
4  Written in 2023 by Jhoanna Olana.
5  */
6  #include <stdio.h>
7  #include <stdbool.h>
8
9  #define NUM_POINTS 9           // Size of the arrays
10 #define POINT_C 2              // Index of charging station C
11 #define POINT_D 3              // Index of charging station D
12
13 int main(void) {
14     // ----- Intialization -----
15     /*
16     Adjacency matrix of the given graph (node/point I included).
17     */
18     bool road_networks[NUM_POINTS][NUM_POINTS] = {
19         [0][0] = 1, [0][1] = 1, [0][5] = 1,
20         [1][0] = 1, [1][1] = 1, [1][2] = 1,
21         [2][1] = 1, [2][2] = 1, [2][4] = 1, [2][5] = 1, [2][8] = 1,
22         [3][3] = 1, [3][4] = 1,
23         [4][3] = 1, [4][4] = 1,
24         [5][0] = 1, [5][2] = 1, [5][5] = 1,
25         [6][0] = 1, [6][3] = 1, [6][6] = 1,
26         [7][7] = 1, [7][8] = 1,
27         [8][7] = 1, [8][8] = 1
28     };
29
30     char points[NUM_POINTS] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'};
31

```

```

32     bool to_C[NUM_POINTS] = {0};
33     bool to_D[NUM_POINTS] = {0};
34
35     int r, c, k, point;
36
37     // ----- Displays the matrix -----
38
39     // Column Header
40     /*
41     Every element in the array named points will be printed as the column header of the adjacency matrix.
42     Charging stations C and D are enclosed in brackets.
43     */
44     printf("\t");
45     for (r = 0; r < NUM_POINTS; r++) {
46         if (points[r] == 'C' || points[r] == 'D') {
47             printf("[%c]\t", points[r]);
48         } else {
49             printf("%c\t", points[r]);
50         }
51     }
52     printf("\n");
53
54     // Row
55     /*
56     Every row in the adjacency matrix consists of the points (act as the headers for the rows)
57     and the edges for each point (or the elements in every row of the array named road_networks).
58     */
59     for (r = 0; r < NUM_POINTS; r++) {
60         // Row Header
61         if (points[r] == 'C' || points[r] == 'D') {
62             printf("[%c]\t", points[r]);
63         } else {
64             printf("%c\t", points[r]);
65         }
66
67         // Elements in each row
68         for (c = 0; c < NUM_POINTS; c++) {
69             printf("%d\t", road_networks[r][c]);
70         }
71
72         printf("\n");
73     }
74     printf("\n");
75
76     // ----- Determines the nearest charging station given a point. -----
77     /*
78     Arrays to_C and to_D tell if there is a path from a certain point to point C/D (point -> C/D = 1).
79     To create these arrays, we iterate through the points(k's) between the given source and the charging stations
80     and test if there is an edge/path from the source to the charging stations through the k's.
81     Example:
82     General Case: If P -> k = 1 and k -> C = 1, then P -> C = 1.
83     Case1 (based on the given): A directly to C = 0, but A -> B = 1 and B -> C = 1. Thus, A -> C = 1.
84     */
85

```



```

86     for (point = 0; point < NUM_POINTS; point++) {
87         for (k = point; k < NUM_POINTS; k++) {
88             if (road_networks[k][POINT_C]) {
89                 to_C[point] = 1;
90                 break;
91             }
92             else if (road_networks[k][POINT_D]) {
93                 to_D[point] = 1;
94                 break;
95             }
96             else {
97                 continue;
98             }
99         }
100         // printf("%c %d %d\n", points[point], to_C[point], to_D[point]);    // for debug
101     }
102
103     /*
104     Do while ensures that users enter a known point only.
105     */
106     do {
107         printf("Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8 - I): ");
108         scanf("%d", &point);
109     } while (point < 0 || point > 8);
110
111     printf("At point: %c\n", points[point]);
112

```

```

113     /*
114     Determines the nearest charging station of a point if that point has a true value in the arrays to_C/D.
115     */
116     if (point == POINT_C || point == POINT_D) {
117         printf("point: %c is a charging station.\n", points[point]);
118     }
119     else if (to_C[point]) {
120         printf("point: C arrived to charging station.\n");
121     }
122     else if (to_D[point]) {
123         printf("point: D arrived to charging station.\n");
124     }
125     else {
126         printf("point: No charging station found.\n");
127     }
128
129     return 0;
130 }

```

```
76 // ----- Determines the nearest charging station given a point. -----
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

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

	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, 8 - I): 0
At point: A
point: C arrived to charging station.

Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8 - I): 1
At point: B
point: C arrived to charging station.

Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8 - I): 2
At point: C
point: C is a charging station.

Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8 - I): 3
At point: D
point: D is a charging station.

Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8 - I): 4
At point: E
point: D arrived to charging station.

Which point are you located? (0 - A, 1 - B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H, 8 - I): 5
At point: F
point: C arrived to charging station.

Github Link: [CMSC21/Lecture4/Assignments at main · jrolana/CMSC21 \(github.com\)](https://github.com/jrolana/CMSC21)