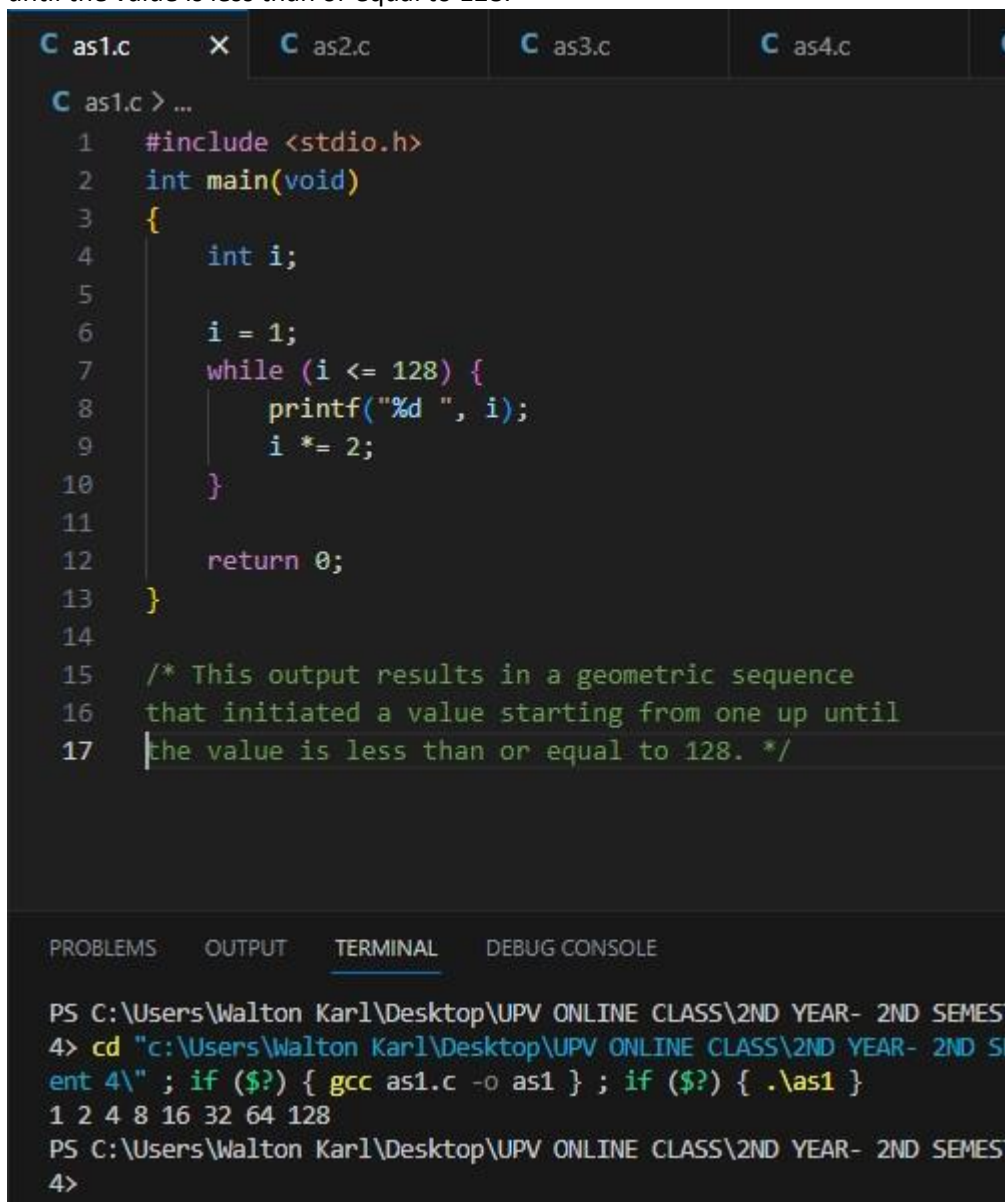1. What is the output of the following program?

```c
#include <stdio.h>

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

Save your code as as1.c

➢ This output results in a geometric sequence that initiated a value starting from one up until the value is less than or equal to 128.

➢

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);
```

Save your code as as2.c
- ➢ The do-while loop is not equivalent to the for loop and the while loop. The while loop and the for loop both evaluates conditions first before executing statements, unlike the do-while loop, which executes the statements first, before evaluating the conditions.

```
C as1.c          C as2.c      ×    C as3.c          C as4.c          C as5.c

C as2.c > ⊘ main(void)
 1    #include <stdio.h>
 2    int main(void)
 3    {
 4        int i;
 5
 6        i = 1;
 7
 8        /* Remove // for the program to work properly */
 9        //while (i < 10) {printf("%d ", i); i *= 2; };
10
11        //for (; i < 10; i *= 2) {printf("%d ", i);};
12
13        //do {printf("%d ", i); i *= 2; } while (i < 10);
14
15        /*
16        The do-while loop is not equivalent to the for loop
17        and the while loop. The while loop and the for loop
18        both evaluates conditions first before executing statements,
19        unlike the do-while loop, which executes the statements first,
20        before evaluating the conditions.
21        */
22
23        return 0;
24    }
25
```
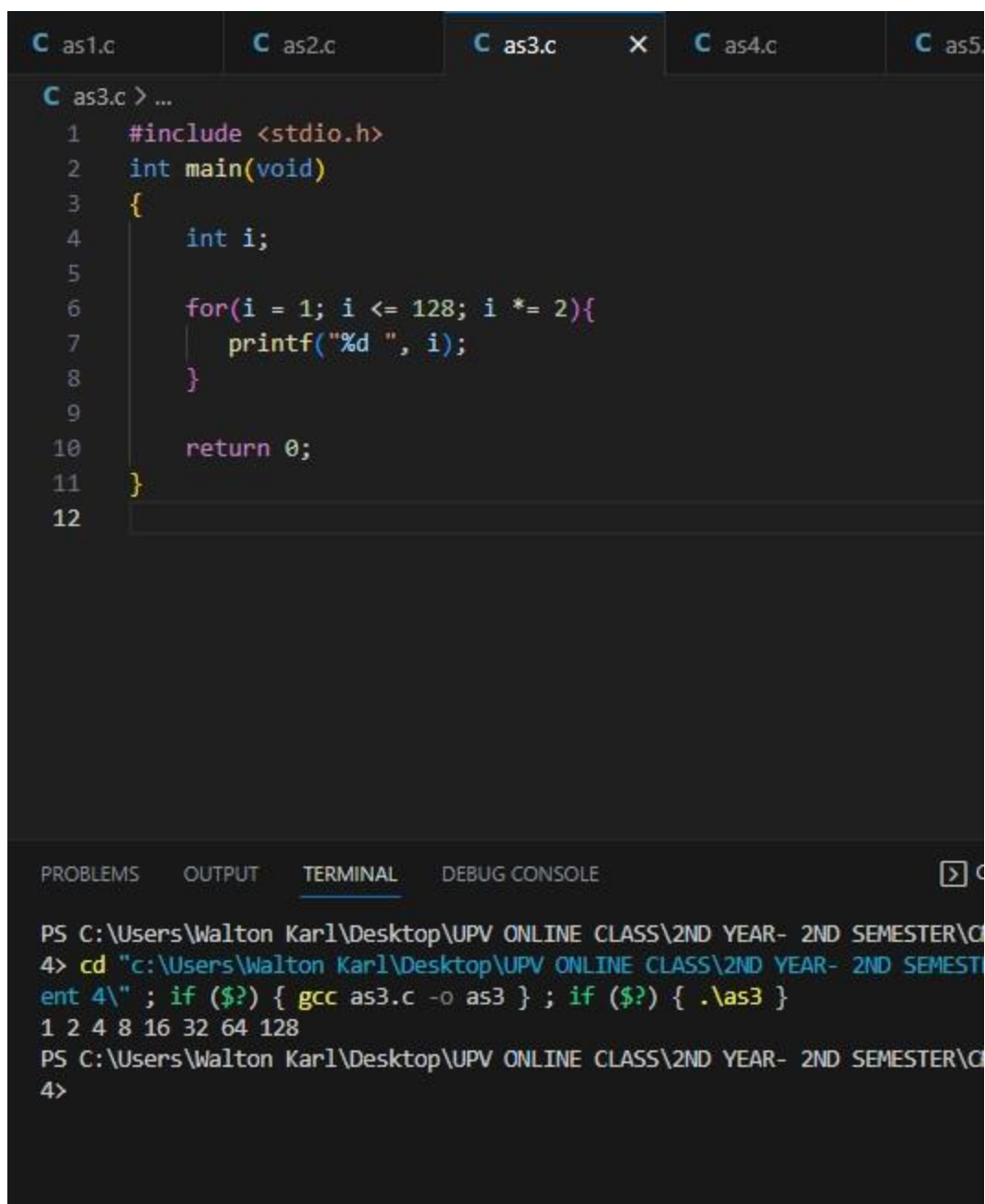
- ➢

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.

Save your code as as3.c

```c
C as1.c          C as2.c          C as3.c   ✕   C as4.c          C as5

C as3.c > ...
  1    #include <stdio.h>
  2    int main(void)
  3    {
  4        int i;
  5
  6        for(i = 1; i <= 128; i *= 2){
  7            printf("%d ", i);
  8        }
  9
 10        return 0;
 11    }
 12
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE   ⊡ ⊂

```
PS C:\Users\Walton Karl\Desktop\UPV ONLINE CLASS\2ND YEAR- 2ND SEMESTER\C
4> cd "c:\Users\Walton Karl\Desktop\UPV ONLINE CLASS\2ND YEAR- 2ND SEMEST
ent 4\" ; if ($?) { gcc as3.c -o as3 } ; if ($?) { .\as3 }
1 2 4 8 16 32 64 128
PS C:\Users\Walton Karl\Desktop\UPV ONLINE CLASS\2ND YEAR- 2ND SEMESTER\C
4>
```

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
10 1024
Save your code as as4.c

```c
#include <stdio.h>
int main(void)
{
    int row = 0, column = 1;

    printf("TABLE OF POWERS OF TWO \n");
    printf("n  2 to the n \n");
    printf("--  -------- \n");

    while(row <= 10){
        while(column <= 1024){
            printf("%d     %d", row, column);
            column += column;
            row++;
            printf("\n");
        }
    }

    return 0;
}
```

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
```
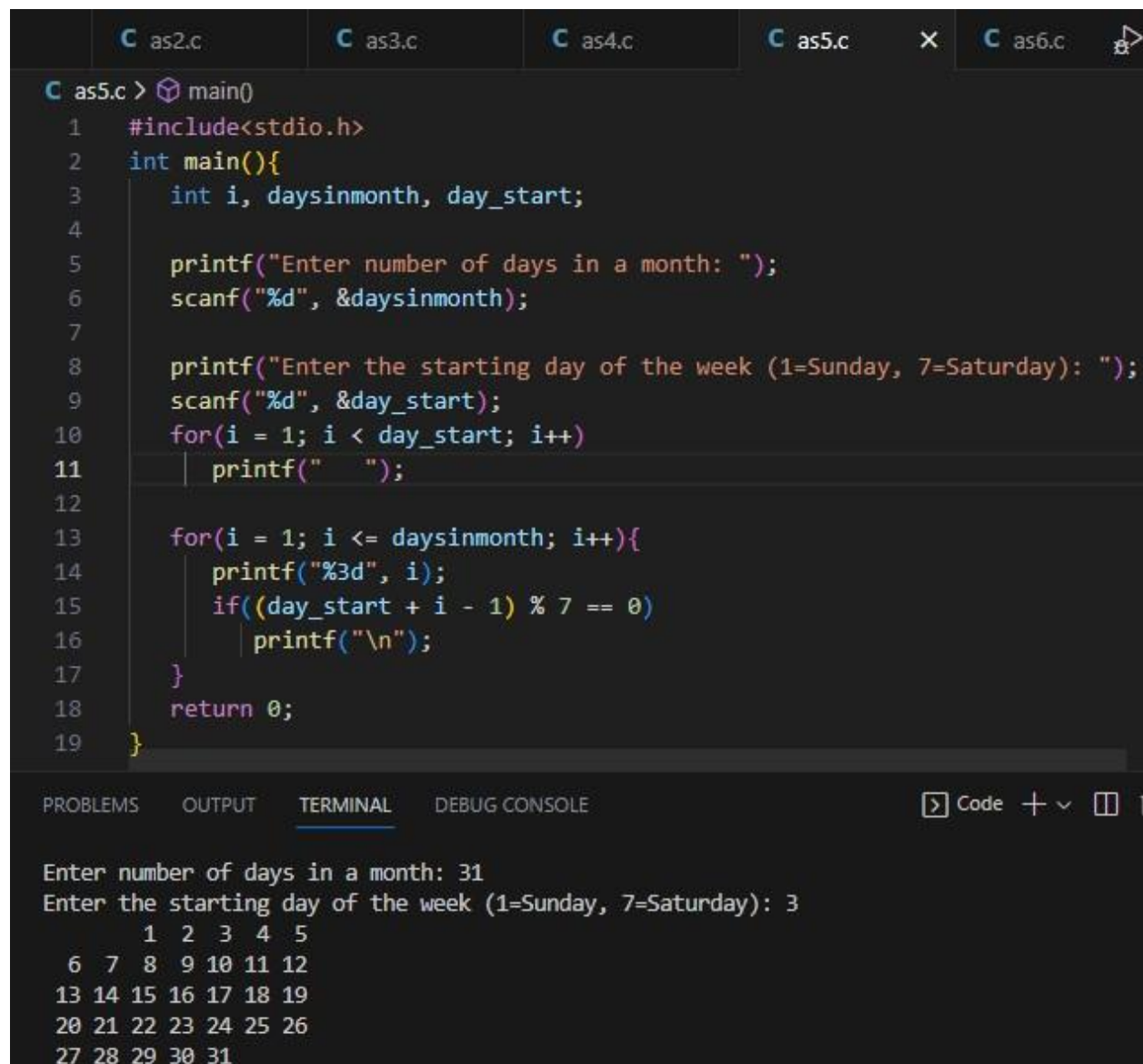
There should be a user prompt to set:
- The number of days
- The day of the week on which the month begins.

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

This addition will be a good refresher to our previous topic, selection statements.

`Save your code as as5.c`

as2.c   as3.c   as4.c   as5.c   ✕   as6.c

C as5.c > ⊗ main()

```c
1    #include<stdio.h>
2    int main(){
3        int i, daysinmonth, day_start;
4
5        printf("Enter number of days in a month: ");
6        scanf("%d", &daysinmonth);
7
8        printf("Enter the starting day of the week (1=Sunday, 7=Saturday): ");
9        scanf("%d", &day_start);
10       for(i = 1; i < day_start; i++)
11           printf("    ");
12
13       for(i = 1; i <= daysinmonth; i++){
14           printf("%3d", i);
15           if((day_start + i - 1) % 7 == 0)
16               printf("\n");
17       }
18       return 0;
19   }
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE   > Code + ∨ ⊓

```
Enter number of days in a month: 31
Enter the starting day of the week (1=Sunday, 7=Saturday): 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. In the program below, an array named pathway contains eight bool values. Each bool element refers to whether a pathway is open or close for transportation.
Only pathways 0 and 2 are open while the rest are still close due to road constructions and fixings.

```c
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)

> ➢ A) Revised line 16:   bool pathway[8] = { [0] = true, [2] = true };
> ➢ B) Revised line 16:   bool pathway[8] = {true, false, true};

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

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

int main(){
    /*

    A boolean array that contains true/false values referring to
    whether a certain pathway is open/close for transportation.

    Only pathways 0 and 3 are open for transportation. The rest are close.

    */

    bool pathway[8] = { [0] = true, [2] = true};

    for(int i = 0; i < NUM_PATHWAYS; i++){

        /*

        Display the status of each pathway.

        Remember that pathway is type bool so its elements are either true/fal

        */

            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
```

```c
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          A boolean array that contains true/false values referring to
10         whether a certain pathway is open/close for transportation.
11
12         Only pathways 0 and 3 are open for transportation. The rest are close.
13
14         */
15
16       bool pathway[8] = {true, false, 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 -
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   }
```

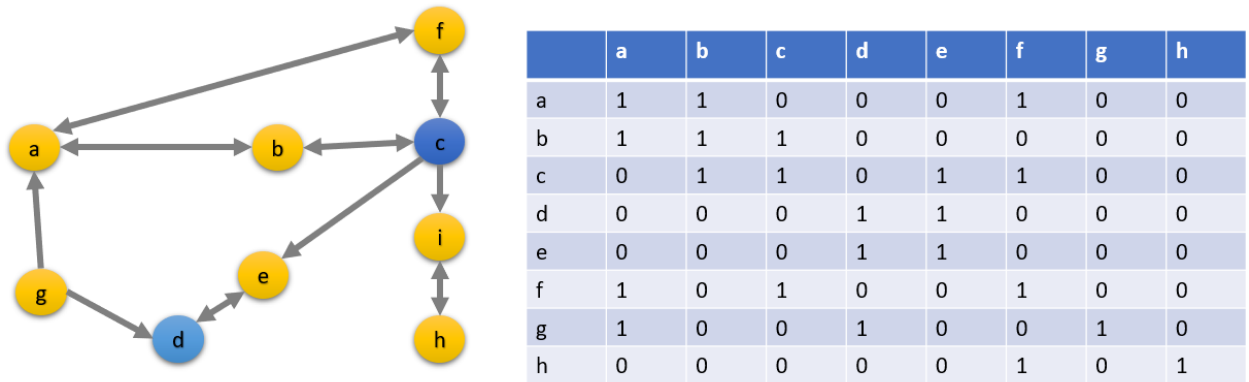PROBLEMS   OUTPUT   **TERMINAL**   DEBUG CONSOLE   ▷ Code + ∨ ⊓ 🗑 ⋯

```
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
```

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

	a	b	c	d	e	f	g	h
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

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

```
38   }
39
40   int main() {
41       int road_networks[NUM_VERTICES][NUM_VERTICES] = {
42           {1, 1, 0, 0, 0, 1, 0, 0},
43           {1, 1, 1, 0, 0, 0, 0, 0},
44           {0, 1, 1, 0, 1, 1, 0, 0},
45           {0, 0, 0, 1, 1, 0, 0, 0},
46           {0, 0, 0, 1, 1, 0, 0, 0},
47           {1, 0, 1, 0, 0, 1, 0, 0},
48           {1, 0, 0, 1, 0, 0, 1, 0},
49           {0, 0, 0, 0, 0, 1, 0, 1}
50       };
51
52       char src;
53       printf("Which point are you located? a, b, c, d, e, f, g, h \n");
54       scanf(" %c", &src);
55       shortestpath(road_networks, src - 'a');
56
57       return 0;
58   }
59
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE   ▷ Code + ∨ ⫿

PS C:\Users\Walton Karl\Desktop\UPV ONLINE CLASS\2ND YEAR- 2ND SEMESTER\CMSC 21\CMSC 21 Ass
"c:\Users\Walton Karl\Desktop\UPV ONLINE CLASS\2ND YEAR- 2ND SEMESTER\CMSC 21\CMSC 21 Assi
if ($?) { gcc as7.c -o as7 } ; if ($?) { .\as7 }
Which point are you located? a, b, c, d, e, f, g, h
e
Nearest charging station for point e: d