

Loops and Arrays

Lecture 4 Assignments

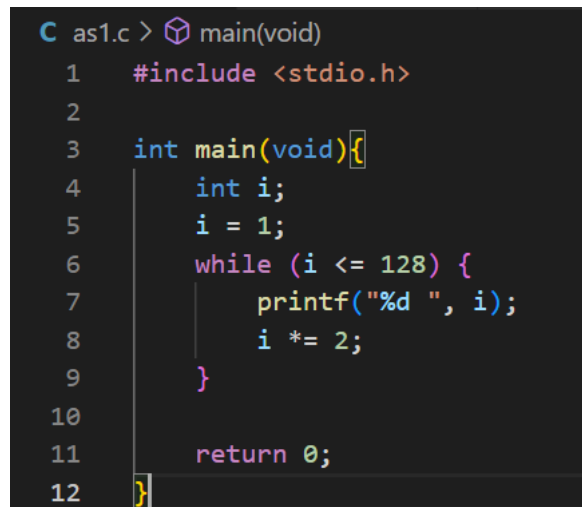
1. What is the output of the following program?

```
#include <stdio.h>

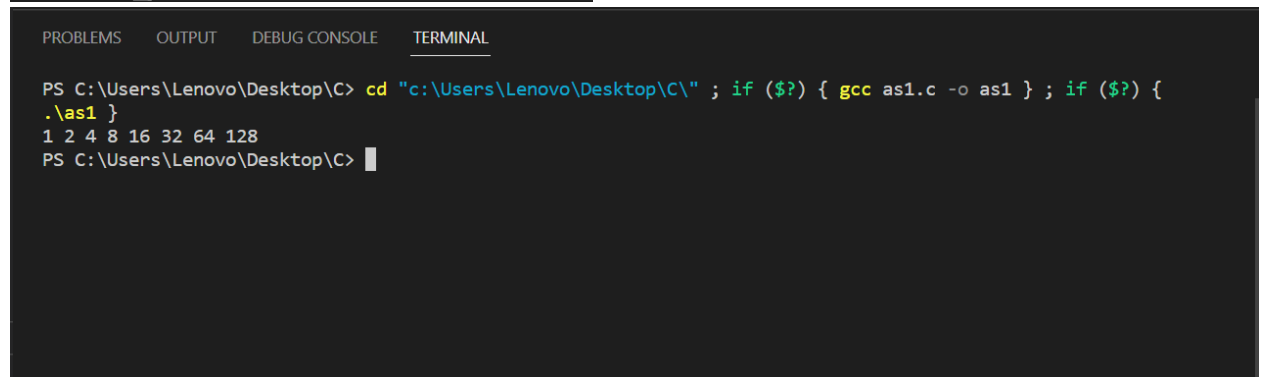
int main(void)
{
    int i;    i = 1;
    while (i <= 128) {
        printf("%d ", i);
        i *= 2;
    }
    return 0;
}
```

Save your code as as1.c

Answer: 1 2 4 8 16 32 64 128

A screenshot of a code editor with a dark background. The code is written in C and is the same as the one in the previous block. The editor shows line numbers from 1 to 12 on the left margin. The code is:

```
C as1.c > main(void)
1  #include <stdio.h>
2
3  int main(void){
4      int i;
5      i = 1;
6      while (i <= 128) {
7          printf("%d ", i);
8          i *= 2;
9      }
10
11     return 0;
12 }
```

A screenshot of a terminal window with a dark background. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal shows the following commands and output:

```
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as1.c -o as1 } ; if ($?) {
.\as1 }
1 2 4 8 16 32 64 128
PS C:\Users\Lenovo\Desktop\C> 
```

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

Answer: c) do {...} while (i < 10); Since a do-while loop will always run at least once.

```
C as2.c > main(void)
1  #include <stdio.h>
2
3  int main(void){
4
5      int i=1;
6
7      for (; i<10; i++){
8          printf("%d\n", i);
9      }
10
11     return 0;
12 }
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as2.c -o as2 } ; if ($?) { .\as2
}
1
2
3
4
5
6
7
8
9
PS C:\Users\Lenovo\Desktop\C>
```

```
C as2.c > main(void)
1  #include <stdio.h>
2
3  int main(void){
4
5      int i=1;
6
7      while(i<10){
8          printf("%d\n", i);
9          i++;
10     }
11
12     return 0;
13 }
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as2.c -o as2 } ; if ($?) { .\as2
}
1
2
3
4
5
6
7
8
9
PS C:\Users\Lenovo\Desktop\C>
```

```
C as2.c > main(void)
1  #include <stdio.h>
2
3  int main(void){
4
5      int i=1;
6
7      do{
8          printf("%d\n", i);
9          i++;
10         }while(i<10);
11
12         return 0;
13     }
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as2.c -o as2 } ; if ($?) { .\as2
}
1
2
3
4
5
6
7
8
9
PS C:\Users\Lenovo\Desktop\C>
```

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 as3.c > ...
1  #include<stdio.h>
2
3  int main(){
4
5      int i;
6
7      for (i = 1; i <= 128; i *= 2)
8      {
9          printf("%d ", i);
10     }
11
12     return 0;
13 }
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as3.c -o as3 } ; if ($?) { .\as3
}
1 2 4 8 16 32 64 128
PS C:\Users\Lenovo\Desktop\C>
```

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 as4.c > main()
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5
6      int i;
7
8      for (i = 0; i <= 10; i++)
9      {
10         printf("2 to the %d is %d. \n", i, (int)pow(2,i));
11     }
12
13     return 0;
14 }
15
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as4.c -o as4 } ; if ($?) { .\as4
}
2 to the 0 is 1.
2 to the 1 is 2.
2 to the 2 is 4.
2 to the 3 is 8.
2 to the 4 is 16.
2 to the 5 is 32.
2 to the 6 is 64.
2 to the 7 is 128.
2 to the 8 is 256.
2 to the 9 is 512.
2 to the 10 is 1024.
```

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`

```
C as5.c > main()
1  #include <stdio.h>
2
3  int main(){
4
5      int i, j, a, b, c=1, A[6][7]={0};
6
7      do{
8          printf("Enter number of days in month:");
9          scanf("%d", &i);
10     }while(i < 28 || i > 31);
11
12     do{
13         printf("Enter the starting day of the week (1=Sun, 7=Sat):");
14         scanf("%d", &j);
15     } while(j < 1 || j > 7);
16
17     if (j == 1)
18     {
19         j = 0;
20     }
21     else
22     {
23         j = j - 1;
24     }
25
26     for (a = 0; a < 6; a++)
27     {
```

```

28     for (b = 0; b < 7; b++)
29     {
30         if (c==1 && b<j)
31         {
32             printf("    ");
33         }
34         else
35         {
36             if (c<=i)
37             {
38                 A[a][b] = c;
39                 printf("%3d ",c);
40                 c++;
41             }
42             else
43             {
44                 break;
45             }
46         }
47     }
48     if (c>i)
49     {
50         break;
51     }
52     printf("\n");
53 }
54
55 return 0;
56 }

```

```

PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as5.c -o as5 } ; if ($?) { .\as5
}
Enter number of days in month:30
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

```

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.

```

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.

```
bool pathway[8] = {[0] = true, [2] = true};
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; 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
```

- b) Revise line 16 such that the initializer will be short as possible (without using a designated initializer)

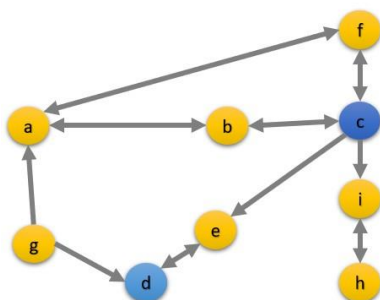
```
bool pathway[8] = {true, false, true};
PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as6.c -o as6 } ; if ($?) { .\as6
}
pathway[0] is open
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

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, $a \rightarrow b = 1$ 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.



	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

```
C as7.c > main()
1  #include <stdio.h>
2
3  #define NUM_PATHWAYS ((int) (sizeof(road_networks) / sizeof(road_networks[0])))
4
5  int main(){
6
7      int i, j, point;
8
9      int road_networks[][9] = {
10         {1, 1, 0, 0, 0, 1, 0, 0, 0},
11         {1, 1, 1, 0, 0, 0, 0, 0, 0},
12         {0, 1, 1, 0, 1, 1, 0, 0, 1},
13         {0, 0, 0, 1, 1, 0, 0, 0, 0},
14         {0, 0, 0, 1, 1, 0, 0, 0, 0},
15         {1, 0, 1, 0, 0, 1, 0, 0, 0},
16         {1, 0, 0, 1, 0, 0, 1, 0, 0},
17         {0, 0, 0, 0, 0, 0, 0, 1, 1},
18         {0, 0, 0, 0, 0, 0, 0, 1, 1}
19     };
20
21     printf("-----ADJACENCY MATRIX-----\n");
22     printf("  A B C D E F G H I\n");
23
24     for (i = 0; i < NUM_PATHWAYS; i++){
25         printf("%c ", 'A' + i);
26         for (j = 0; j < NUM_PATHWAYS; j++){
27             if (i == 2 || i == 3 || j == 2 || j == 3){
28                 printf("[%d]", road_networks[i][j]);
29             }else{
```



```

30         printf("%2d ", road_networks[i][j]);
31     }
32 }
33     printf("\n");
34 }
35     printf("\n");
36
37     printf("Which point are you located? 0-A, 1-B, 2-C, 3-D, 4-E, 5-F, 6-G, 7-H, 8-I: ");
38     scanf("%d", &point);
39     printf("\nAt point: %c", 'A' + point);
40
41     switch (point){
42         case 0: case 1: case 5:
43             printf("\nPoint: C arrived to charging station");
44             break;
45         case 4: case 6:
46             printf("\nPoint: D arrived to charging station");
47             break;
48         case 2:
49             printf("\nPoint: C is a charging station");
50             break;
51         case 3:
52             printf("\nPoint: D is a charging station");
53             break;
54         default:
55             printf("\nPoint: No charging station reached");
56             break;
57     }
58
59     return 0;
60 }

```

```

PS C:\Users\Lenovo\Desktop\C> cd "c:\Users\Lenovo\Desktop\C\" ; if ($?) { gcc as7.c -o as7 } ; if ($?) { .\as7
}
-----ADJACENCY MATRIX-----
  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: 5

At point: F
Point: C arrived to charging station
PS C:\Users\Lenovo\Desktop\C>

```

Instructions for submissions

- Take screenshots of your codes for numbers which requires coding (e.g., 1, 2, 3) and embed it on the pdf along with an example output.
- Submit your answers in a pdf file with filename assignment2[surname].pdf
- Save the pdf file (assignment4[surname].pdf) and the codes in the directory: CMSC21/Lecture4/Assignments/
- Remember that you have initially created this repository for your reading assignment. • Upload to github.
- Download git cmd
- Navigate to the CMSC21 Folder
- For example (assuming your CMSC21 folder is in Documents)
- `cd Documents/CMSC21`
- `git add -all`
- `git commit -m "Lecture 4 Assignment"`
- `git push -u origin main`
- Email me the github link with subject:
[CMSC 21] Assignment 4 (Surname) [Date Submitted]