

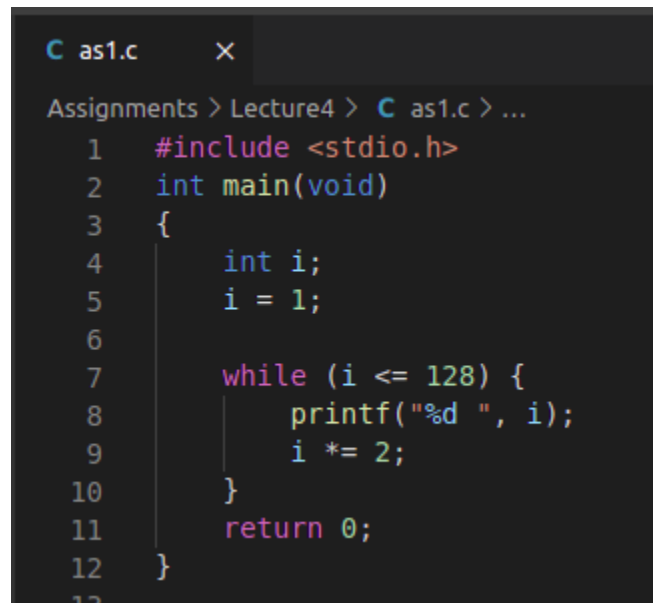
Ellabelle E. Garcia
1st Year Computer Science
Ara Abigail E. Ambita
CMSC 21
4/5/22

Loop/Repetition Statements 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

A screenshot of a code editor window titled "C as1.c". The editor shows the same C code as in the previous block. The code is:

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

Output: 1 2 4 8 16 32 64 128

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

- Letter C which is the do-while statement is not equivalent to the other two. The do-while loop is most often used when a program wants to do something at least once before checking the condition because in the while loop and for loop, the block of codes will not get executed even once if the condition is not satisfied.

Output

```
While Loop
1 2 3 4 5 6 7 8 9
For Loop
1 2 3 4 5 6 7 8 9
Do-while Loop
10
```

```
C as2.c x
Assignments > Lecture4 > C as2.c > ...
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i;
6      i = 1;
7
8      printf("While Loop\n");
9      while (i < 10) {
10         printf("%d ", i);
11         i++;
12     }
13
14     printf("\nFor Loop\n");
15     for (i = 1; i < 10; i++)
16     {
17         printf("%d ", i);
18     }
19
20     printf("\nDo-while Loop\n");
21     do
22     {
23         printf("%d \n", i);
24         i++;
25     } while (i < 10);
26
27
28     return 0;
29 }
```

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

Output: 1 2 4 8 16 32 64 128

```
C as3.c x
Assignments > Lecture4 > 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      {
8          printf("%d ", i);
9      }
10
11     return 0;
12 }
```

4. Write a code that computes for the power of two

```
C as4.c x
Assignments > Lecture4 > C as4.c > ...
1  #include<stdio.h>
2
3  int main(void)
4  {
5      int i, pow_n;
6
7      printf("\nTABLE OF POWERS OF TWO");
8      printf("\n=====");
9      printf("\n n      2 to power n");
10     printf("\n-----");
11     for (i = 0; i <= 10; ++i)
12     {
13         if (i == 0)
14         {
15             pow_n = 1;
16         }
17         else
18         {
19             pow_n *= 2;
20         }
21         printf("\n %d\t %d", i, pow_n);
22     }
23     printf("\n=====");
24
25     return 0;
26 }
```

Output

TABLE OF POWERS OF TWO	
=====	
n	2 to power n

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	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  X
Assignments > Lecture4 > C as5.c > ...
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int days, first_day, week_day, day_count;
6      char retry;
7
8      do
9      {
10         printf("\n***** One-Month Calendar Generator *****");
11         printf("\nEnter number of days in month: ");
12         scanf("%d", &days);
13
14         // check if the entered value for the number of days is valid
15         if (days > 31 || days < 28)
16         {
17             printf("Input error: The entered number of days is invalid!\n");
18         }
19         else
20         {
21             printf("Enter starting day of the week (1=Sun, 7=Sat): ");
22             scanf("%d", &first_day);
23             printf("\n");
24
25             // check if the entered value for starting day of the week is valid
26             if (first_day > 7 || first_day < 1)
27             {
28                 printf("Input error: The input should be from 1 to 7! \n");
29             }
30             else
31             {
32                 printf(" One-Month Calendar\n-----\n");
33                 printf(" Su Mo Tu We Th Fr Sa\n");
34                 /* printing the blank days of the first week */
35                 for (week_day = 1; week_day < first_day; week_day++)
36                 {
37                     printf("  ");
38                 }
39
40                 /* printing the calendar numbers */
41                 for (day_count = 1; day_count <= days; week_day++, day_count++)
42                 {
43                     printf("%3d", day_count);
44                     if (week_day % 7 == 0)
45                         printf("\n");
46                 }
47                 printf("\n\n");
48             }
49         }
50         /* Asks the user to try again */
51         printf("\nDo you want to try again? (y or n): ");
52         scanf(" %c", &retry);
53     } while (retry == 'y' || retry == 'Y');
54
55     printf("Thank you for using this program!\n");
56
57     return 0;
58 }

```

github link: <https://github.com/ellabellegarcia/CMSC21/tree/main/Lecture4/Assignments>