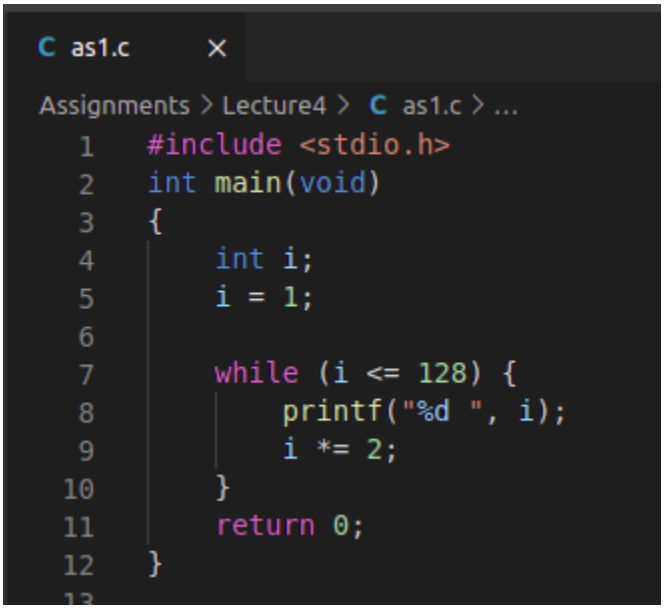
Ellabelle E. Garcia
1st Year Computer Science
Ara Abigail E. Ambita
CMSC 21
4/5/22

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



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
```

as2.c

Assignments > Lecture4 > C as2.c > ...

```c
#include <stdio.h>

int main(void)
{
    int i;
    i = 1;

    printf("While Loop\n");
    while (i < 10) {
        printf("%d ", i);
        i++;
    }

    printf("\nFor Loop\n");
    for (i = 1; i < 10; i++)
    {
        printf("%d ", i);
    }

    printf("\nDo-while Loop\n");
    do
    {
        printf("%d \n", i);
        i++;
    } while (i < 10);


    return 0;
}
```

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

as3.c

Assignments > Lecture4 > C as3.c > ...

```c
#include <stdio.h>
int main(void)
{
    int i;

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

    return 0;
}
```

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

```c
#include<stdio.h>

int main(void)
{
    int i, pow_n;

    printf("\nTABLE OF POWERS OF TWO");
    printf("\n=====================");
    printf("\n  n     2 to power n");
    printf("\n-----  ---------------");
    for (i = 0; i <= 10;  ++i)
    {
        if (i == 0)
        {
            pow_n = 1;
        }
        else
        {
            pow_n *= 2;
        }
        printf("\n  %d\t    %d", i, pow_n);
    }
    printf("\n=====================\n");

    return 0;
}
```

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
```

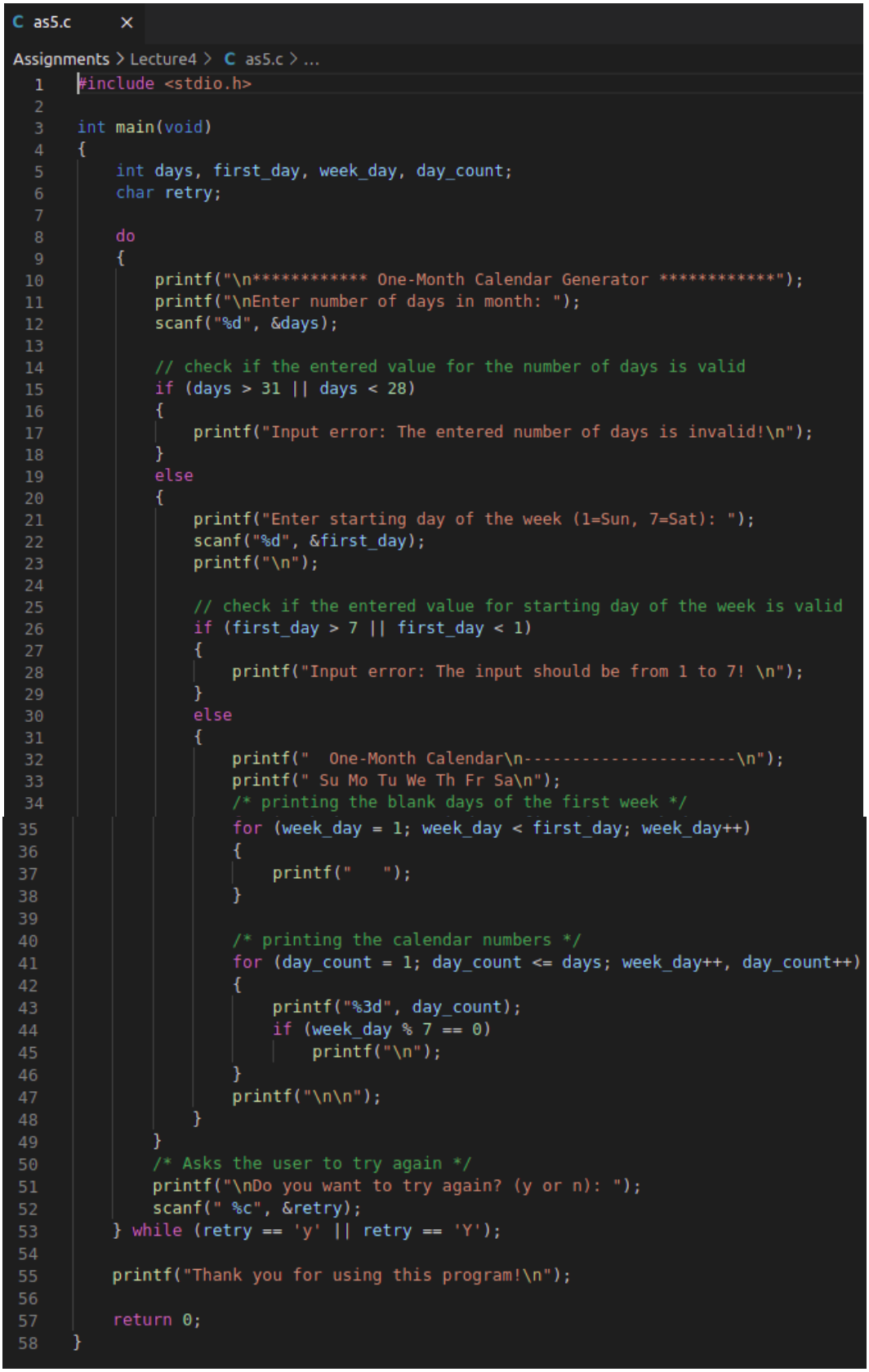
There should be a user prompt to set:

- 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`



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