Cordero, Palmsdale Kevin D. pdcordero@up.edu.ph
2021-03842

Loop/Repetition Statements
Lecture 4 Assignment

1.

```c
/* ~~~~~~~~~~~~~~~~~~~~~~~~~
Cordero, Palmsdale Kevin D.
2021-03842
pdcordero@up.edu.ph
~~~~~~~~~~~~~~~~~~~~~~~~~ */

#include <stdio.h>

int main(void) {
    int i;

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

    return 0;
}
```

```
1 2 4 8 16 32 64 128
```

Output:
1 2 4 8 16 32 64 128

2.

```c
/* ~~~~~~~~~~~~~~~~~~~~~~~~~
Cordero, Palmsdale Kevin D.
2021-03842
pdcordero@up.edu.ph
~~~~~~~~~~~~~~~~~~~~~~~~~ */

#include <stdio.h>

int main(void) {
    int i, inp;

    printf("Input i: ");
    scanf("%d", &inp);

    printf("WHILE\n");
    i = inp;
    while (i<10) {
        printf("%d ", i);
        i++;
    }

    i = inp;
    printf("\nFOR\n");
    for (; i<10;) {
        printf("%d ", i);
        i++;
    }

    i = inp;
    printf("\nDO WHILE\n");
    do {
        printf("%d ", i);
        i++;
    } while (i<10);

    return 0;
}
```

```
Input i: 0
WHILE
0 1 2 3 4 5 6 7 8 9
FOR
0 1 2 3 4 5 6 7 8 9
DO WHILE
0 1 2 3 4 5 6 7 8 9
```

```
Input i: 15
WHILE

FOR

DO WHILE
15
```

All statements are equivalent whenever `(i<10)` is true. However, statement c, which is a do-while loop, deviates whenever `(i<10)` is false. This is because the do-while loop will still execute its body even if the condition was not satisfied, while the other statements will never execute their bodies. Thus, statement c is the outlier of the three.

3.

```c
/* ~~~~~~~~~~~~~~~~~~~~~~~~
Cordero, Palmsdale Kevin D.
2021-03842
pdcordero@up.edu.ph
~~~~~~~~~~~~~~~~~~~~~~~~ */

#include <stdio.h>

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

```
1 2 4 8 16 32 64 128
```

4.

```c
/* ~~~~~~~~~~~~~~~~~~~~~~~~
Cordero, Palmsdale Kevin D.
2021-03842
pdcordero@up.edu.ph
~~~~~~~~~~~~~~~~~~~~~~~~ */

#include <stdio.h>

int main(void) {
    int inp, x, y;
    // x = n
    // y = 2^n

    printf("Input n: ");
    scanf("%d", &inp);

    printf("-n- ---2^n---\n");
    y = 1;
    for (x = 0; x<=inp; x++) {
        printf("%3d %-9d\n", x, y);
        y*=2;
    }

    return 0;
}
```

```
Input n: 25
-n- ---2^n---
  0 1
  1 2
  2 4
  3 8
  4 16
  5 32
  6 64
  7 128
  8 256
  9 512
 10 1024
 11 2048
 12 4096
 13 8192
 14 16384
 15 32768
 16 65536
 17 131072
 18 262144
 19 524288
 20 1048576
 21 2097152
 22 4194304
 23 8388608
 24 16777216
 25 33554432
```

```
Input n: 0
-n- ---2^n---
  0 1
```

5.

```c
/* ~~~~~~~~~~~~~~~~~~~~~~~~
Cordero, Palmsdale Kevin D.
2021-03842
pdcordero@up.edu.ph
~~~~~~~~~~~~~~~~~~~~~~~~ */

#include <stdio.h>

int valid_input(int min, int max) {
    /* Ensures input is within validity range from
    given minimum to given maximum */
    int inp;

    while(1) { // exits only when inp is within range
        printf("\nInput [%d-%d]: ", min, max);
        scanf("%d", &inp);

        if (inp>=min && inp<=max) {     // valid, within range
            return inp;
        } else {                        // invalid
            printf("Invalid input! Must be from %d to %d only. Try again.", min, max);
            // loop back to Input
        }
    }
}
```

```c
int main(void) {
    int days, start, n;

    /*~~~ INPUTS ~~~*/
    printf("How many days in the month?");
    days = valid_input(28, 31);
    printf("\nWhich day of the week to start on?\n    1.) Sunday\n    2.) Monday\n    3.) Tuesday\n    4.) Wednesday\n    5.) Thursday\n    6.) Friday\n    7.) Saturday");
    start = valid_input(1, 7);

    /*~~~ OUTPUT CALENDAR ~~~*/
    printf("\nSu Mo Tu We Th Fr Sa\n");

    // Start of Week Spacing
    for (n=1; n<start; n++) {
        printf("   ");
    }

    // Days
    start = (8-start)%7; // signifies end of row

    for (n=1; n<=days; n++) {
        printf("%2d ", n);

        // when end of row, print new line
        if (n%7==start) {
            printf("\n");
        }
    }

    return 0;
}
```

```
How many days in the month?
Input [28-31]: 0
Invalid input! Must be from 28 to 31 only. Try again.
Input [28-31]: 27
Invalid input! Must be from 28 to 31 only. Try again.
Input [28-31]: 32
Invalid input! Must be from 28 to 31 only. Try again.
Input [28-31]: 29
```

```
Which day of the week to start on?
    1.) Sunday
    2.) Monday
    3.) Tuesday
    4.) Wednesday
    5.) Thursday
    6.) Friday
    7.) Saturday
Input [1-7]: 0
Invalid input! Must be from 1 to 7 only. Try again.
Input [1-7]: 8
Invalid input! Must be from 1 to 7 only. Try again.
Input [1-7]: 5
```

```
Su Mo Tu We Th Fr Sa
                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
```

```
How many days in the month?
Input [28-31]: 28

Which day of the week to start on?
    1.) Sunday
    2.) Monday
    3.) Tuesday
    4.) Wednesday
    5.) Thursday
    6.) Friday
    7.) Saturday
Input [1-7]: 6
```

```
Su Mo Tu We Th Fr Sa
                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
```

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