Judelle Clareese E. Gaza
CMSC 21 -1

1.

```c
// Judelle Gaza

#include <stdio.h>

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

**OUTPUT:**

```
PS C:\Users\ACER> cd "c
 .\as1 }
1 2 4 8 16 32 64 128
```

1 2 4 8 16 32 64 128

2.

```c
// Judelle Gaza

#include <stdio.h>

int main(void){
    int i, input;

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

    // WHILE - prints i while i is less than 10
    printf("a) while\n");
    i = input;
    while (i<10){
        printf("%d ", i);
        i++;            }   // increments i

    // FOR 0 - prints i while i is less than 10
    i = input;
    printf("\nb) for\n");
    for (;i<10;) {
        printf("%d ", i);
        i++;            }   // increments i

    // DO WHILE - executes 'do' even if while condition is not satisfied
    i = input;
    printf("\nc) do while\n");
    do{
        printf("%d ", i);
        i++;            }   // increments i
    while (i<10);

    return 0;
}
```

**OUTPUT:**

**i.**

```
Input i: 1
a) while
1 2 3 4 5 6 7 8 9
b) for
1 2 3 4 5 6 7 8 9
c) do while
1 2 3 4 5 6 7 8 9
```

**ii.**

```
Input i: 11
a) while

b) for

c) do while
11
```

The while and for loops only execute when the given condition ( i < 10 ) is true, on the other hand the do – while loop will still execute its body even if the given condition is not satisfied. Statement c is the outlier of the 3.

3.

```c
// Judelle Gaza

#include <stdio.h>

int main(void) {
    int i;

    /*  equivalent for loop of number 1
        initiate i as 1,
        the condition i < = 128,
        and the after execution or the updation of i after executing body */
    for (i = 1 ;i <= 128; i*=2){
        printf("%d ", i);
    }
    return 0;
}
```

**PRINTS THE SAME OUTPUT AS 1:**

1 2 4 8 16 32 64 128

```
PS C:\Users\ACER> cd "c
1 2 4 8 16 32 64 128
PS C:\Users\ACER\OneDri
```

4.

```c
// Judelle Gaza

#include <stdio.h>

int main(void) {
    // Declaring Variables
    int input, n, e;

    // Asks and Scans for user input
    printf("Input n: ");
    scanf("%d", &input);

    // Prints out the table of 2 to the nth power
    printf("  n    2^n   \n============\n");
    e = 1;
    for (n = 0; n <= input; n++) {              // n starts at 0, condition while n is <= than the user input, and n is incremented for every iteration
        printf("%3d     %-9d\n", n, e);
        e *= 2;                         }       // e is multiplied by 2 n times while the for loop is executed
    return 0;
}
```

**OUTPUT:**

```
Input n: 10
  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
```

5.

```c
// Judelle Gaza

#include <stdio.h>

int main(){
    int days, startday, n;

    // While loop for continuous user input if input is Invalid for the number of days
    while (1){
        printf("How many days in a Month?\nInput [28-31]: ");
        scanf("%d", &days);
        // breaks if condition is satisfied
        if (days >= 28 && days <= 31) {
            break;                    }
        else{
            printf("Invalid Input. Please Try Again. ");}}

    // While loop for continuous user input if input is Invalid for the starting day
    while (1){
        printf("\nWhich day of the week to start? [1] Sunday [2] Monday [3] Tuesday [4] Wednesday [5] Thursday [6] Friday [7] Saturday\nInput [1-7]: ");
        scanf("%d", &startday);
        // breaks if condition is satisfied
        if (startday >= 1 && startday <= 7) {
            break;                         }
        else{
            printf("Invalid Input. Please Try Again. ");}}

    // CALENDAR PRINTING
    printf("\n=== YOUR CALENDAR ===\nSu Mo Tu We Th Fr Sa\n");

    /* Prints Spaces if n is less than startday
        - if the starting day is not on a sunday */
    for (n=1; n < startday; n++){
        printf("   ");          }

    /* Prints the days of the month calendar - use %2d to indicate the spacing
        increments n and prints n until the said number of days, also increments n*/
    for (n=1; n <= days; n++)       {
        printf("%2d ", n);
        // prints to a new line every end of the row
        if (n%7 == (8-startday) % 7){
            printf("\n");          }}
    return 0;
}
```

OUTPUT:

```
How many days in a Month?
Input [28-31]: 31

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

=== YOUR CALENDAR ===
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 30 31
```

```
How many days in a Month?
Input [28-31]: 28

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

=== YOUR CALENDAR ===
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
```

```
How many days in a Month?
Input [28-31]: -1
Invalid Input. Please Try Again. How many days in a Month?
Input [28-31]: 0
Invalid Input. Please Try Again. How many days in a Month?
Input [28-31]: 27
Invalid Input. Please Try Again. How many days in a Month?
Input [28-31]: 32
Invalid Input. Please Try Again. How many days in a Month?
Input [28-31]: 30

Which day of the week to start? [1] Sunday [2] Monday [3] Tuesday [4] Wednesday [5] Thursday [6] Friday [7] Saturday
Input [1-7]: 8
Invalid Input. Please Try Again.
Which day of the week to start? [1] Sunday [2] Monday [3] Tuesday [4] Wednesday [5] Thursday [6] Friday [7] Saturday
Input [1-7]:
```