1. Output: 1 2 4 8 16 32 64 128

```c
#include <stdio.h>
int main(void)
{
 int i;
 i = 1;
 while (i <= 128) {
 printf("%d ", i);
 i *= 2;
 }
 return 0;
}
//output: 1 2 4 8 16 32 64 128
```

```
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>exec1
1 2 4 8 16 32 64 128
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>
```

2.

```c
/*

a) while (i < 10) {...};
b) for (; i < 10;) {...};
c) do {...} while (i < 10);

a and b are the same.
c, by definition, will test the condition after the code has executed at least once. A and b meanwhile test before the code execute

*/
```

A and B are the same functionally since a for loop is just a while loop without it first and third statements.
The third statement has to check its statement after the code has executed once, while the other two have to check before the code is executed.

3.

```c
#include <stdio.h>
int main(void)
{
 int i;
 i = 1;
 for (; i <= 128;) {
 printf("%d ", i);
 i *= 2;
 }
 return 0;
}
```

```
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>exec3
1 2 4 8 16 32 64 128
```

4.

```c
#include <stdio.h>
int main(void)
{
    int exponent = 0, final = 1;
    printf("n | 2 to the n");
    while (exponent <= 10) {
        printf("\n%d | %d", exponent, final);
        final *=2;
        exponent += 1;
    }
    return 0;
}
```

```
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>exec4
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
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>
```

5.

```c
#include <stdio.h>
int main(void)
{
    const int DAYSPERROW = 7;
    int numOfDays, startDay, daysInRow = 1, dayIndex = 1;
    printf("Enter number of days in month (no going over 31!): ");
    scanf("%d", &numOfDays);
    if (numOfDays <= 0 || numOfDays >= 32)
    {
        printf("You entered an invalid number!");
        return 0;
    }
    printf("Enter starting day of the week (1=Sun, 7=Sat): ");
    scanf("%d", &startDay);
    if (startDay <= 0 || startDay >= 7)
    {
        printf("You entered an invalid number!");
        return 0;
    }

    for (int i = 1; i <= startDay - 1; i++)
    {
        printf("  "); //2 spaces for two digits, other 2 spaces for space in between days of week.
        printf("  ");
        daysInRow++;
        if (daysInRow > 7)
        {
            printf("\n");
            daysInRow = 1;
        }
    }

    for (int i = 1; i <= numOfDays; i++)
    {
        printf("%d", dayIndex); //2 spaces for two digits, other 2 spaces for space in between days of week.
        printf("  ");
        dayIndex++;
        daysInRow++;
        if (daysInRow > 7)
```

```c
    printf("Enter starting day of the week (1=Sun, 7=Sat): ");
    scanf("%d", &startDay);
    if (startDay <= 0 || startDay >= 7)
    {
        printf("You entered an invalid number!");
        return 0;
    }

    for (int i = 1; i <= startDay - 1; i++)
    {
        printf("  "); //2 spaces for two digits, other 2 spaces for space in between days of week.
        printf("  ");
        daysInRow++;
        if (daysInRow > 7)
        {
            printf("\n");
            daysInRow = 1;
        }
    }

    for (int i = 1; i <= numOfDays; i++)
    {
        printf("%d", dayIndex); //2 spaces for two digits, other 2 spaces for space in between days of week.
        printf("  ");
        dayIndex++;
        daysInRow++;
        if (daysInRow > 7)
        {
            printf("\n");
            daysInRow = 1;
        }
    }
    return 0;
}
```

```
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>exec5
Enter number of days in month (no going over 31!): 10
Enter starting day of the week (1=Sun, 7=Sat): 2
    1  2  3  4  5  6
7  8  9  10
C:\Users\bonif\Documents\Activities\CMSC21\Lecture4\Assignments>
```