1) Determine the output and/or identify the error in each of the following code segments:

a)

```
let r = 0;
while (r < 10); {
    console.log ("r is " + r);
    r = r + 3;
}
```

Output/ Error:

b)

```
let s = 11;
do {
    s = s - 2;
    console.log ("s is " + s);
} while (s >= 1);
```

Output/ Error:

c)

```
let t = 0, u = 4;
while (t <= 3) {
    t++;
    for (let v = t; v < u; v++) {
        console.log (v + t);
    }
}
```

Output/ Error:

d)

```
let m = 10, n = 4;
while (m > 5) {
   if (m % n != 1)
        console.log(m);
   m = m - 1;
}
```

Output/ Error:

e)

```
let msg = "";
for (let d = 0; d < 5; d ++) {
    if (d % 2 == 1) {
        for (let e = 1; e <= d; e++) {
            msg = msg + e;
        }
    }
    msg = msg + "\n";
    for (let f = 6; f > 2; f = f-2;) {
        msg = msg + f;
    }
}
console.log(msg);
```

Output/ Error:

2) Using the while loop, write the program to generate each of the following output:

a)

```
Output:
6 12 18 24 30
```

b)

```
Output:
80-40-20-10-5
```

3) Using the do-while loop, write the program to generate each of the following output:

a)

```
Output:
2 4 8 16 32 64
```

b)

```
Output:
91 + 80 - 69 + 58 - 47
```

You may open up your Visual Studio Code (VSC) and work from there for the remaining questions.

[Optional : After you have completed each question, try challenging yourself to include some data validation in your program. Do remember to prepare test cases to ensure your validation works for all possible cases.]

4) Write a **do-while** loop to prompt for the number of children from the user until a valid input in the range of 0 to 50 is captured. An appropriate error message should be displayed for invalid input.

Create a Test plan showing all possible test cases used to test your program.

Sample output:

```
Please enter the number of children: -5
Invalid number of children!
Please enter in the range of 0 to 50.

Please enter the number of children: 51
Invalid number of children!
Please enter in the range of 0 to 50.

Please enter the number of children: 4
Input Accepted! Program terminated...
```

5) Write a program using **do-while** loop that prompts user to enter a number. The number entered must be between 100 and 200 (both inclusive). If the number entered is not within the range, it displays an error message and prompts user to re-enter.

Create a Test plan showing all possible test cases used to test your program.

Sample output:

```
Enter a number: 50
Error! Please enter a number between 100 and 200.

Enter a number: 245
Error! Please enter a number between 100 and 200.

Enter a number: 123
Input Accepted: End of Program!
```

6) Write a program to handle the following requirements:

- Prompts the user to enter an **integer**
- Display an appropriate message if the number is divisible by 3 but not 5.
- The user will be prompted repeatedly to enter the next number until the user enters 0 as the input.
- Create a Test plan showing all possible test cases used to test your program.

Sample output:

```
Enter any number or (0) to quit: 12
12 is divisible by 3 but not 5? true

Enter any number or (0) to quit: 30
30 is divisible by 3 but not 5? false

Enter any number or (0) to quit: 0
Program Terminated…
```

[Optional Questions]

7) Modify question 6, such that users are only allowed to enter a number that are between 50 and 100 (inclusive) or 0 to quit the program.

Recap the use of Math.abs in Practical 3b, Q5.

Sample output:

```
Enter any number or (0) to quit: 188
Error: Out of range!

Enter any number or (0) to quit: 44
Error: Out of range!

Enter any number or (0) to quit: 63
63 is divisible by 3 but not 5? true

Enter any number or (0) to quit: 55
55 is divisible by 3 but not 5? false

Enter any number or (0) to quit: 0
Program Terminated…
```

8) Write a program that produces a multiplication table with 25 rows of computations. Allow the user to input the first and last base values for the multiplication table.

Display a column in the table beginning with the first base inputted value. The last column should be the ending base value entered. The first row should be for 1 times the beginning base, 1 times the (beginning base value + 1), through 1 times the ending base value. The last row should be for 25 times the beginning base, 25 times the (beginning base value + 1), through 25 times the ending base value. Base values can range from 2 through 7.

Display an aesthetically formatted multiplication table. An example of output produced when 2 and 7 are entered appears in the figure below.

Sample output:

```
Enter first base value: 2
Enter last base value: 7
n       2       3       4       5       6       7
1       2       3       4       5       6       7
2       4       6       8       10      12      14
3       6       9       12      15      18      21
4       8       12      16      20      24      28
5       10      15      20      25      30      35
6       12      18      24      30      36      42
7       14      21      28      35      42      49
8       16      24      32      40      48      56
9       18      27      36      45      54      63
10      20      30      40      50      60      70
11      22      33      44      55      66      77
12      24      36      48      60      72      84
13      26      39      52      65      78      91
14      28      42      56      70      84      98
15      30      45      60      75      90      105
16      32      48      64      80      96      112
17      34      51      68      85      102     119
18      36      54      72      90      108     126
19      38      57      76      95      114     133
20      40      60      80      100     120     140
21      42      63      84      105     126     147
22      44      66      88      110     132     154
23      46      69      92      115     138     161
24      48      72      96      120     144     168
25      50      75      100     125     150     175
```

- END -