Points 4: choose one of two options

Option 1

1. Declare a numerical variable and set value to 7.
2. Print zeros in size n×n like this (1 point):

```
0000000
0000000
0000000
0000000
0000000
0000000
0000000
```

3. Print diagonal using **+** across zeros (1 point):

```
+000000
0+00000
00+0000
000+000
0000+00
00000+0
000000+
```

4. Print borders using **=** (1 point)

```
=======
=+0000=
=0+000=
=00+00=
=000+0=
=0000+=
=======
```

Rule: your solution must work with different values of n (we assume n >= 1), i.e. n=20:

```
====================
=+0000000000000000=
=0+0000000000000000=
=00+000000000000000=
=000+00000000000000=
=0000+0000000000000=
=00000+000000000000=
=000000+00000000000=
=0000000+0000000000=
=00000000+000000000=
=000000000+00000000=
=0000000000+0000000=
=00000000000+000000=
=000000000000+00000=
=0000000000000+0000=
=00000000000000+000=
=000000000000000+00=
=0000000000000000+0=
=00000000000000000+=
====================
```

Option 2

1. Declare two numerical variables: x and y (i.e. with values: 8 and 4)
2. Print x rows of y columns where:
    a. First element is row number
    b. Each next element in column is a next power of a column before

Example:

```
1   1   1   1
2   4   8   16
3   9   27  81
4   16  64  256
5   25  125 625
6   36  216 1296
7   49  343 2401
8   64  512 4096
```

You can use Math.pow(x,y)