

## Assignment 1:

## Loops

1. Create a list of 20 random elements, with values ranging from 1 to 100. Store all the multiples of 2, all the multiples of 3 and all the multiples of 6 in separate lists. If a number is a multiple of 6, it shouldn't appear in the multiples of 2 or multiples of 3 lists. Print out all three multiples lists, as well as the list of remaining elements, that are not a multiple of any of the three. Output the results in ascending order. Feel free to use the inbuilt sorting functions. For example, my output looks like:

```
C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>python q1_multiples.py
Multiples of 2: [28, 28, 46, 56, 58, 62]
Multiples of 3: [75]
Multiples of 6: [6, 30, 66, 84, 90]
Remainders: [7, 7, 37, 41, 49, 61, 67, 95]

C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>
```

**[IS 615 only]** Modify the code so that there can be no duplicates in the random 20 elements. For example, in the screenshot above, there shouldn't be double 7s. Your code should print out the list of duplicate elements found (sorted in ascending order):

```
C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>python q1_multiples_615.py
Multiples of 2: [4, 44, 46, 50, 76, 100]
Multiples of 3: [27, 45]
Multiples of 6: [12, 24, 30, 54]
Remainders: [7, 17, 49, 61, 65, 73, 83, 95]
Duplicates: [12, 65]

C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>
```

2. Create a table of powers. The program should calculate the square, cube and fourth power of the numbers from 1 to N, where N is provided on the command line as an input argument. Your code should check that the input argument has been provided, and ensure that it is an integer greater than or equal to 1. You don't need to deal with the possibility of the input not being an integer (text for example). Make sure that your output table is padded dynamically, so that there is always at least 2 spaces between each column. For example, my output looks like (N=4):

```
C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>python q2_powers.py 4
i  i^2  i^3  i^4
1   1   1   1
2   4   8  16
3   9  27  81
4  16  64 256

C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>
```

And (N=6)

```
C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>python q2_powers.py 6
i  i^2  i^3  i^4
1   1   1   1
2   4   8  16
3   9  27  81
4  16  64 256
5  25 125 625
6  36 216 1296

C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>
```

**[IS 615 only]** Modify the code so that instead of providing the maximum value to calculate the square, cube and fourth power for (N), the user provides the upper limit of the fourth power calculation. For example, if the user provides the number 256, then the program will only calculate up to N=4, because  $4*4*4*4 = 256$ . But if the user provides 9000, then the program calculates up to a value of N=9, because  $9*9*9*9 = 6561$  and  $10*10*10*10 = 10000$ , which is too big:

```
C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>python q2_powers_615.py 9000
i  i^2  i^3  i^4
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
9  81 729 6561

C:\Users\wcox1\Box\UNR\2023 Spring\IS 415\Assignments\Assignment1 - Loops>
```