Introduction to programming – ASSIGNMENTS 4

General instructions:

- Type in and test your programs using Python Idle.
- Bring in your answers to the next assignment (i.e. demonstration) session in memory stick, or save them to a web folder accessible in class. Alternatively, you can bring written answers with you, though this is not recommended.
- Remember to comment your code – this does NOT mean, that every single line should be commented. Comment the important parts in your program.
- Prepare to present your solution to the class.

1.

Write a program that queries the user for a number N, and then proceeds to output all numbers between (-N and N).

Example output:

```
Enter a number: 3
-3
-2
-1
0
1
2
3
```

2.

Write a program that queries the user for number, and proceeds to output whether the number is an even or an odd number. Program terminates, when the user enters a zero.

Example run:

```
Enter a number or a zero to quit: 8
That's an even number
Enter a number or a zero to quit: 13
That's an odd number
Enter a number or a zero to quit: 0
Bye!
```

3.

Write a program that queries the user for a number N, and the proceeds to output all powers of two smaller than N. For example, if N is given a value 9, numbers 1, 2, 4 and 8 are output (2^0, 2^1, 2^2 and 2^3).

Example run:

```
Enter a number: 21
1
2
4
8
16
```

4.

Write a program that queries the user for a string, and proceeds to generate a new string with all of the letters from the original string that are in upper case. This generated string is then output. You can assume that the string given does not contain characters other than letters (e.g. numbers or spaces).

Example run:

```
Enter a string: AbcDeFgYTkm
Characters in upper case: ADFYT
```

5.

String B is said to be a *permutation* of string A, if an equal amount of each character in A can be found in B and vice versa. Hence, B is said to be a *rearrangement* of A. For example, strings “ABBA”, “BAAB” and “ABAB” are all permutations of string “AABB”, and string “elvis” and “sevil” are permutations of string “lives”. Note, that these string permutations are sometimes called *anagrams*.

Write a program that queries the user for strings A and B, and proceeds to output whether B is a permutation of A. To confirm this, the following conditions need to be met:

1. There is an equal amount of characters in A and B (or, the length of A is equal to length of B), and
2. For every character found in A, there is an equal amount of same character in B.

Example run:

Enter string A: ABCD
Enter string B: DBAC
DBAC is a permutation of ABCD.

6. ** *Expert assignment (double points)*

A *prime number* is a number that has exactly two natural number (number > 0) divisors: one and the number itself. Hence, the ten smallest prime numbers are 1, 2, 3, 5, 7, 11, 13, 17, 19 and 23.

Write a program that queries the user for a number and proceeds to output whether the number is a prime number or not.

Example run:

```
Enter a number: 43
43 is a prime number
```

Another example run:

```
Enter a number: 22
22 is not a prime number
```