Assignment 1.1: Insertion Sort (3 points)

Following pseudo code sorts an array of integers. Command swap switches values of two variables.

```
function isort(A)
    for i = 1 to size(A)-1
        j = i-1
        while (j >= 0) and (A[j] > A[j+1])
            swap(A[j], A[j+1])
            j = j-1
```

Create the following function in Python:

- `isort(A: list)`: sorts a given list of integers using the insertion sort. Implement function based on the above pseudo code. Note that the use of the Python's built-in sort() method is not allowed.

Limits (your code should work with an input that is within these limits):

- the maximum length of the list is 10^3
- each integer is between $1 \ldots 10^3$

A code template with an example program:

```
def isort(A):
    # TODO


if __name__ == "__main__":
    A = [4, 3, 6, 2, 9, 7, 1, 8, 5]
    isort(A)
    print(A)
```

Output:

```
$ python isort.py
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Submit your solution in CodeGrade as `isort.py`.

Assignment 1.2 Prime Numbers (3 points)

For the background read the first paragraph of this article: https://en.wikipedia.org/wiki/Prime_number

Given a number N how many prime numbers are less or equal to N? For example if $N = 7$ there are four prime numbers: 2, 3, 5 and 7 (note that 1 is not a prime number).

Create the following function in Python:

- `primes(N: int)`: returns the numbers of primes that are less or equal to N

Limits: $1 \leq N \leq 10^5$

A code template with an example program:

```
def primes(N):
    # TODO


if __name__ == "__main__":
    print(primes(7))
    print(primes(15))
    print(primes(50))
```

Output:

```
$ python primes.py
4
6
15
```

Submit your solution in CodeGrade as `primes.py`.

Assignment 1.3: Is it a Triangle? (3 points)

Three integers a, b and c presents the side lenghts of a triangle. Can you build any triangle from those three sides?

For example:

1. sides 3, 4 and 5 makes a right angle triangle
2. sides 5, 5 and 3 makes an isosceles triangle
3. sides 7, 3 and 3 doesn't make a triangle
4. sides 4, 3 and -1 doesn't make a triangle

Create the following function in Python:

- `triangle(a: int, b: int, c: int)`: returns a boolean `True` if triangle can be built, `False` if not

A code template with an example program:

```
if __name__ == "__main__":
    print(triangle(3, 5, 4))
    print(triangle(-1, 2, 3))
    print(triangle(5, 9, 14))
    print(triangle(30, 12, 29))
```

Output:

```
$ python triangle.py
True
False
False
True
```

Submit your solution in CodeGrade as `triangle.py`.

Last modified: Wednesday, 4 September 2024, 2:16 PM