National University of Singapore
School of Computing
CS1010S: Programming Methodology
Semester I, 2024/2025

Tutorial 6
Sequences & Lambda

Release date: 7th October 2024
Due: 13th October 2024, 23:59

General Restrictions

- No importing packages unless explicitly allowed to do so.
- Use only `tuple` as your compound data structure. No `list`, `set`, `dict` etc.

Questions

1. Implement a Python function called **`odd_indices`** that takes in a tuple as its only argument and returns a tuple containing all the elements with odd indices (i.e. every second element from the left) from the input tuple. For example:

```
>>> odd_indices(('a', 'x', 'b', 'y', 'c', 'x', 'd', 'p', 'q'))
('x', 'y', 'x', 'p')
```

2. Implement a function called **`even_odd_sums`** that takes in a tuple of numbers as its only argument and returns a tuple of two elements: the first is the sum of all even-indexed numbers in the input tuple, while the second element is the sum of all odd-indexed elements in the input tuple.
Sample execution:

```
>>> even_odd_sums((1, 3, 2, 4, 5))
(8, 7)
>>> even_odd_sums((1,))
(1, 0)
>>> even_odd_sums(())
(0, 0)
```

3. Suppose `x` is bound to the tuple `(1, 2, 3, 4, 5, 6, 7)`. Using `map`, `filter` and/or lambdas (as discussed in Lecture), write an expression involving `x` that returns:
   (a) `(1, 4, 9, 16, 25, 36, 49)`
   (b) `(1, 3, 5, 7)`
   (c) `((1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7))`
   (d) `(4, 8, 16, 20, 28)`
   (e) `((3, 1), (9, 27), (15, 125), (21, 343))`
   (f) `(1, 8, 9, 64, 25, 216, 49)`

   You are encouraged to provide multiple solutions for the above questions.