

National University of Singapore
School of Computing
CS1010X: Programming Methodology
Semester II, 2024/2025

Tutorial 5
Working with Sequences

1. Draw box-and-pointer diagrams for the values of the following tuples.

`((1, 2, (3,)), (4, 5), (6, 7))`

`(1, (2, 3, (4,)))`

`(1, (2, (3, (4, 5))))`

2. Write expressions using index notation that will return the value 1 when the identifier `tup` is bound to the following values:

`(7, (6, 5, 4), 3, (2, 1))`

`((7), (6, 5, 4), (3, 2), 1)`

`(7, (6,)), (5, (4,)), (3, (2, (1,)))`

`(7, ((6, 5), (4,)), 3, 2), ((1,)),)`

Note that index notation can be chained. For example:

```
>>> tup = (7, (6, 5, 4), 3, (2, 1))
>>> tup[1][2]
4
```

3. Write a Python function called `even_rank` that takes in a tuple as its only argument and returns a tuple containing all the elements of even rank (i.e. every second element from the left) from the input tuple. For example:

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

4. Write a function called `odd_even_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 odd-ranked numbers in the input tuple, whereas the second element is the sum of all even-ranked elements in the input. Example execution:

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

```
>>> odd_even_sums((1, ))
(1, 0)
```

```
>>> odd_even_sums(())
(0, 0)
```

5. In this question, you will be required to build a variant of the solution to the “Towers of Hanoi” problem presented in class. We define a *disk move* to be a pair of two numbers: the source pole and the destination pole. For example, (1, 3) indicates the move of a disk from the first pole to the third.

Write a function called `hanoi` that takes in 4 parameters:

- the number of disks,
- the source pole.
- the destination pole,
- the auxiliary pole,

and *returns a tuple of disk moves* that, if executed in that sequence, will move all the disks from the source pole to the destination pole and comply with the rules of the Tower of Hanoi game. (Hint: you will not get any marks for a solution that prints a sequence of moves, since that has already been given in class).

Example execution:

```
>>> hanoi(1, 1, 2, 3)
((1, 2),)
```

```
>>> hanoi(1, 1, 3, 2)
((1, 3),)
```

```
>>> hanoi(3, 1, 2, 3)
((1, 2), (1, 3), (2, 3), (1, 2), (3, 1), (3, 2), (1, 2))
```