Weekly Lab

Search Algorithms

In this lab session, we will experiment with number of search algorithms.

Exercise 1

Write a program to determine **the first found position** of an element in the one-dimensional list of integers with different search algorithms:

1. Linear search (implement with while).
2. Linear search with sentinel (implement with while).
3. Binary search (the input list is sorted in ascending order already).
4. (Optional) Interpolation search (the input list is sorted in ascending order already).

Using command line: `$python Exercise_1.py algorithm x input_path output_path`
For example: `$python Exercise_1.py 1 3 input.txt output.txt`
where:

- `algorithm` represents the algorithm used, e.g., `1` is Linear search algorithm.
- `x` is the value that need to be determined position, e.g., `3` is search key.
- `input_path` represents the path to the input file, e.g., `input.txt`.
- `output_path` represents the path to the output file, e.g., `output.txt`.

Input format:

- The first line contains a single integer N, which is the size of list.
- The next line denotes the list's elements. Each element is separated by a space.

For example:

```
6
2 8 3 9 1 3
```

Output format:
The first found position of x in the list (position count from 0). If not found, output contains `-1`.
For example: `2`

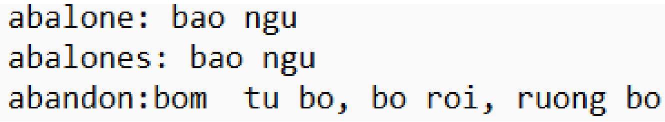
Exercise 2

Build `WORD` structure to store word in language with name and its definition. Apply any search algorithm to create a dictionary software.

Using command line: `$python Exercise_2.py word1 word2 word3 ... output.txt`

- `word1`, `word2` and so on are words which user want to know definition.
- `output.txt` is output file which contains their definitions. Each word and its definition is displayed similar with dictionary file.

In the dictionary, each line will store one word and its definitions. Word and definition are separated by the colon. The following figure is an example of dictionary.



```
abalone: bao ngu
abalones: bao ngu
abandon:bom  tu bo, bo roi, ruong bo
```

Figure 1: Dictionary file.

Regulations

Please follow these guidelines:

- You may use any Python IDE.
- After completing assignment, check your submission before and after uploading to Moodle.
- Do not use the following modules: `numpy`, `pandas`, `collections`, `heapq`, and `deque`.
- You may use `list`, `tuple`, and `set` but no external libraries.

Your submission must be contributed in a compressed file, named in the format `StudentID.zip`, with the following structure:

```
StudentID
├── Exercise_1.py
└── Exercise_2.py
```

The end.