Group Assignment 1 - The Vault Runner (Supplement)
AI1030 - Python Programming

1 Example

In this section, we give a very simple example of another real-life scenario referring to interacting with a dog. To keep it simple, we assume that we have the following two commands: BARK NTIMES (i.e., bark N times) and JUMP. An example program in this language is the following:

```
BARK 2
JUMP
BARK 1
```

Next, let us describe how to implement the Interpreter for this language. As explained in the assignment brief, the input program should be given as a list of strings at the beginning of the Interpreter code (for convenience and simplicity only).

```python
myprogram = [  # Each element of the list is a line of code
    "BARK 2",
    "JUMP",
    "BARK 1"
]

numlines = len(myprogram) # number of lines of input program
counter = 0 # the line being executed by my interpreter

# Loop until the end of program.
while counter < numlines:
    line = program[counter].strip()
    if not line:  # if the line is empty to to the next one
        counter += 1
        continue

    parts = line.split()  # Split the line of code in tokens.

    if parts[0] == "BARK": # BARK command identified
        times = int(parts[1])
        for _ in range(times):
            print("Dog barks!")
        counter += 1

    elif parts[0] == "JUMP": # JUMP command identified
        print("Dog jumps!")
        counter += 1

    else:  # Error detection.
        print(f"Syntax error: {line}")
```

Listing 1: An example implementation of the Interpreter in Python.

After running the previous Python program, the output is as follows:

```
Dog barks!
Dog barks!
Dog jumps!
Dog barks!
```

2 General Guidelines

- Respect the rules by not changing the non-negotiable elements of the assignment.

- Try to design a simple language. The more complex your language, the more difficult the implementation of the interpreter will be.

- First, assume that your robot navigation program is correct without syntax errors. Try to catch syntax errors at the second stage.

- Try to write clear code with comments wherever needed. The only data structure that you really need is the list and it is used only to store the input program. You will not need any advanced list manipulation techniques (more about lists in Week 3).