Release date: 2022/08/17 Deadline: 2022/08/31 23:00

Problem Statement

IIT Delhi has recently deployed a drone for aerial surveillance on account of independence day. The drone begins at position `(0,0,0)` and can move infinitely in any direction depending on how it is programmed. A drone program is a string consisting of a sequence instructions, where each instruction is one of the following.

- `+X`: Move one unit in the direction of the positive X-axis.
- `-X`: Move one unit in the direction of the negative X-axis.
- `+Y`: Move one unit in the direction of the positive Y-axis.
- `-Y`: Move one unit in the direction of the negative Y-axis.
- `+Z`: Move one unit in the direction of the positive Z-axis.
- `-Z`: Move one unit in the direction of the negative X-axis.
- m`(P)`, where $m > 0$ is an integer and P is a drone program: Execute program `P` m times.

For example,

- `2(+X+Y-Z)` is equivalent to `+X+Y-Z+X+Y-Z`, moving the drone to $(2, 2, -2)$ after traveling distance 6.
- `5(+X)10(-X)` is equivalent to `+X+X+X+X+X-X-X-X-X-X-X-X-X-X-X`, moving the drone to $(-5, 0, 0)$ after traveling distance 15.
- `3(-Y2(+Z))` is equivalent `-Y+Z+Z-Y+Z+Z-Y+Z+Z`, moving the drone to $(0, -3, 6)$ after traveling distance 9.
- `+X+X+X+X4(+Y)2(+Z-Z)` is equivalent to `+X+X+X+X+Y+Y+Y+Y+Z-Z+Z-Z`, moving the drone to $(4, 4, 0)$ after traveling distance 12.

Your task is to write a **python** program that takes a drone program `P` as input and outputs the following.

1. The final position of the drone after it has executed its program `P`.
2. The total distance travelled by the drone in this process.

To solve this problem, it will be helpful to use stacks, so **implement the `Stack` data structure and its member functions from scratch**. For full credit, **your program must run in time $O(n)$ on drone programs of length n**. The length of a drone program is the number of characters in the program. For example, the length of the program `9999(+X)` is 8 (and not 9999). We will assume that your program runs in linear time provided it terminates within a specific amount of time set into our auto-grader. To minimise the possibility of a false auto-grader timeout, you are advised to remove all unnecessary print statements that you might have written to debug your program.

Submission Specifications

Submit a single file named `a1.py`. Your submitted file must contain a function `findPositionandDistance(P)` that takes a string `P` as input, and return a list `[x,y,z,d]` containing four numbers, where (x, y, z) is the final position of the drone after it executes program `P`, and d is the distance traveled by the drone in the process.

Example Test Cases

```
>>> findPositionandDistance('+X+Y+X-Y-Z+X+X-Z-Z-Z-Z-Y+Y-X')
[3, 0, -5, 14]
>>> findPositionandDistance('+X2(+Y-X-Z)8(+Y)9(-Z-Z)')
[-1, 10, -20, 33]
>>> findPositionandDistance('')
[0, 0, 0, 0]
>>> findPositionandDistance('5(2(3(+X+X)))')
[60, 0, 0, 60]
>>> findPositionandDistance('+Z6(+X+Y+X-Y)9(-X+Z-X-Z8(+X+Y-Z)9(+Y-Z-X-Y4(-X+Y-X-Z+X)))')
[-339, 396, -476, 2221]
>>> findPositionandDistance('1(+X)5(+Y)41(+Z)1805(-X)3263441(-Y)10650056950805(-Z)')
[-1804, -3263436, -10650056950764, 10650060216098]
```