Please follow the instructions **STRICTLY,** otherwise, you will get **ZERO** score !!!

Please follow the instructions **STRICTLY,** otherwise, you will get **ZERO** score !!!

Please follow the instructions **STRICTLY,** otherwise, you will get **ZERO** score !!!

Overall rules

- Store your code for each question in **a single python file**
  - Code for Question 1 should be stored in `q1.py` file
  - Code for Question 2 should be stored in `q2.py` file
  - Code for Question 3 should be stored in `q3.py` file
  - **Case sensitive** !!!
    - Q1.py, Q2.py, Q3.py are **NOT** acceptable
- Put these three files in a folder which is named with your **student ID**
  - *e.g.,*
    - |-- 123000000/
      - |-- q1.py
      - |-- q2.py
      - |-- q3.py
- Zip the **folder** in **a zip file**
  - *e.g.*, `123000000.zip`
  - Put the python files in a folder first, and then **zip the folder**!
- Submit **ONLY** the `.zip` file on BlackBoard.
- A template is uploaded on BlackBoard. You cna follow it.

Question 1

Instructions:

- `q1.py` : the file containing the class
- The class is named as `Flower` (case sensitive)
- The order of the input for the class
  - name: `str`
  - number of petals: `int`
  - price: `float`

- The `Flower` class should contain a method named `getValue`
  - `getValue` : return a `dict` variable with three key-value pairs
    - 1st key-value pair: `str:str`
    - 2nd key-value pair: `str:int`
    - 3rd key-value pair: `str:float`

Example of how we will test your program

```
from q1 import *
flower = Flower('juice',5,7.82)
flower.getValue()
```

The `flower.getValue()` should return a `dict` type variable with value `{"name": "juice", "numPetal": 5, "price": 7.82 }` .

Your code should be robust enough, for example,
If an instance is initiallized using:

```
flower = Flower(52,5,7.82)
```

it should print "The input of the name is incorrect. A string is required".

If an instance is initiallized using:

```
flower = Flower('juice',5.2,7.82)
```

it should print "The input of the number of petals is incorrect. An integer is required".

If an instance is initiallized using:

```
flower = Flower('juice',5,7)
```

or

```
flower = Flower('juice',5,'seven')
```

it should print "The input of the price is incorrect. A float is required".

If an instance is initiallized using:

```
flower = Flower(52,5.2,'seven')
```

only the first type error will be reported,
i.e., it should print "The input of the name is incorrect. A string is required".

Notice

- You do NOT have to ask the user to re-input if they input an incorrect value.
  - If you already write this function, it's also okay.

Qestion 2

Instructions:

- `q2.py` : the file containing the class
- The class is named as `Polynomial` (case sensitive)
- The order of the input for the class is a `str` variable representing a polynomial
  - e.g., `"2*x^3+3*x^2+5*x+1"`
- The `Polynomial` class should contain a `derivative` method
  - `derivative` : return a `str` variable representing the first-order derivative of the original polynomial

Example of how we will test your program

```
from q2 import *
polynomial = Polynomial("2*x^3+3*x^2+5*x+1")
polynomial.derivative()
```

The `polynomial.derivative()` should return a `str` type variable with value `"6*x^2+6*x+5"`

Your code should handle the following situation:

- The inputted polynomial will contain only one variable, and the variable is an English letter that is not necessarily ‘x’;
  - *e.g.*, If an instance is initiallized using:
    - `polynomial = Polynomial("2*y^3+3*y^2+5*y+1")`
  - then `polynomial.derivative()` should return `"6*y^2+6*y+5"`
- In the inputted polynomial, the terms are not necessarily arranged in descending or scending orders, and the output simply follow the order of **the original polynomial**
  - *e.g.*, If an instance is initiallized using:
    - `polynomial = Polynomial("3*y^2+2*y^3+1+5*y")`
  - then `polynomial.derivative()` should return `"6*y+6*y^2+5"`
  - If an instance is initiallized using:
    - `polynomial = Polynomial("3*y^2+5+4*y^3+5*y")`
  - then `polynomial.derivative()` should return `"6*y+12*y^2+5"`

The testing will follow the following rules, so you do **NOT** have to worry the robustness:

- The degree of each term of the inputted polynomial must be a non-negative integer
- In the inputted polynomial, the coefficients will be placed before the variable
- In the inputted polynomial, no two terms will have an identical degree.

Notice

- The consideration of negative coefficient will be treated as bonus
  - which means if you lose some points somewhere else in Q1 or Q2, it can make it up
- All the coefficient will be separated from the variable with a `*`
- The coefficients are all integer
- We won't test a case involving multiple invoking
  - *e.g.*, `polynomial.derivative().derivative()`

Question 3

In charge by Li Wang

1 Overview

This assignment is about simulating the movement of fish and bears in an ecosystem. You have been provided with a template code, and here are the instructions on how to complete this assignment.

2 Tasks

You need to modify the template code to accomplish the following tasks:

1. Randomly initialize a certain number of fish and bears in the river.
2. Simulate the movement of fish and bears in the river, with each move limited to adjacent positions.

3 Template Code

In the template code, you need to fill in two places to complete the tasks:

1. Initialize the positions of fish and bears in the river.
2. Write code to implement the random movement of animals.

Please note that, apart from the two places mentioned above, do not modify the code outside of the TODO comments. The code will print the state of the river after each step, and the grading will be based on the printed results.

In the template, I provide 4 questions, each worth 10% points.

4 Note

The initial state of the river must be printed. For example, the test 1, the initial river state is *FNB*, after first step, may be *FBN*, after second step, may be *BNN*. Therefore, the print is:

FNB
FBN
BNN

If a fish or bear moves out of the river's boundaries, it will no longer exist in the ecosystem.

Newly generated species will not move, but will be eaten or form new animals with the preceding animal.

Please remember to run the four use cases provided by the template in your submitted code.

1. The length of the river is 3, the number of fish is 1, and the number of bears is 1. Simulate 2 steps. (Print three lines of results, the first line shows the initial river state, and the second and third lines show the river state after each simulation.)

2. The length of the river is 10, the number of fish is 3, and the number of bears is 2. Simulate 5 steps. (Print six lines of results, the first line shows the initial river state, and the second to sixth lines show the river state after each simulation.)

3. The length of the river is 8, the number of fish is 2, and the number of bears is 1. Simulate 10 steps. (Print eleven lines of results, the first line shows the initial river state, and the second to eleventh lines show the river state after each simulation.)

4. The length of the river is 10, the number of fish is 2, and the number of bears is 3. Simulate 10 steps. (Print eleven lines of results, the first line shows the initial river state, and the second to eleventh lines show the river state after each simulation.)

5 Submitting the Assignment

Please submit the modified code along with any additional explanations or comments as your assignment. You can check the state of the river after each move to ensure that the code runs as expected.

Good luck with your assignment! If you have any questions, feel free to ask.