IDBS Coding Assessment Exercise- Python

This document describes how to complete the Assessment Exercise using Python.

All data for the exercise is available within the text files provided.

Polynomial Level 1

Levels of Difficulty

There are two levels of difficulty for this question (you are currently working to level-1):

- Level 1 - simplest level (basic input and calculations)
- Level 2 - medium complexity (level-1 + varied input sources)

The general scenario and rules described below (level-1 rules) apply across all two levels of difficulty for this question. Each level of difficulty adds additional rules and builds on the previous and is designed to give you the opportunity to demonstrate your design, technical and coding skills.

The level of difficulty that you decide to deliver in your final solution is entirely up to you - but remember we are looking for high quality design and coding skills.

Scenario & Rules: Level-1

In this question you'll be doing a mathematical calculation, with a coding twist. You need to evaluate multiple polynomial equations, i.e. for a given polynomial equation you'll calculate the value of y for a given value of x.

The polynomial will have the following form:

$$\mathbf{y = 3x^4 + 2x^3 - 6x^2 + 8x - 2}$$

- the number of terms in the polynomial will not exceed 10 terms
- the value of the multiplier will not exceed 10
- the value of X will not exceed 50
- all numbers used (the power, multiplier, constant and value of X) will always be positive integers

Now here is the coding twist:

The code that you write to evaluate the equation
must ONLY utilise the mathematical operators
ADD and SUBTRACT.

You **are not allowed** to use mathematical operators such as multiplication, division, modulas, "raised to the power of", logarithms or any mathematical functions/operators other than ADD or SUBTRACT. If the code you submit uses any method other than simple ADD or SUBTRACT during the calculation process the solution will be rejected.

Use of the increment/decrement operator ++ or -- (for example: i++ as a loop counter) is allowed within the code of your solution, but the value should not be used as part of an addition or subtraction within the polynomial calculation itself.

Level-1 Specifics:
The polynomial will be provided as input data in the form of Json as follows, (this json is for the example above):

```
{
   "xValue": 5,
   "terms":
      [
         {
            "power": 1,
            "multiplier": 8,
            "action": "add"
         },
         {
            "power": 0,
            "multiplier": 2,
            "action": "subtract"
         },
         {
            "power": 4,
            "multiplier": 3,
            "action": "add"
         },
         {
            "power": 3,
            "multiplier": 2,
            "action": "add"
         },
         {
            "power": 2,
            "multiplier": 6,
            "action": "subtract"
         }
      ]
}
```

Testing Your Solution

The data representing a single polynomial is provided on each line of the input file:

LEVEL_1_<date/time>.txt

The expected answer for a given equation is provided on the following line of the file (i.e. the value of y).

Each file for Level_1 contains a set of 100 polynomials with each polynomials being randomly generated according to the criteria specified above.

Your task is to read the file content, calculate the value of y for each input string (according the rules defined above) and compare this to the expected value – ensuring you have calculated the value correctly.

Once you have all test cases passing AND you have coded to your chosen difficulty level AND you are satisfied with the quality of code you have written - **you are ready to submit your solution** - please see ***General Info*** for instructions on submitting your solution.

What We're Looking For

In this assessment exercise we are assessing your design and coding skills so your code should be of the highest quality – well designed using OOD/OOP practices, readable, reliable, robust and maintainable. Please use any techniques that you feel appropriate for such high quality code and maintainability.

If you write any supporting unit test code, please feel free to submit this to us as part of the assessment (you MAY use mathematical operators other than ADD and SUBTRACT in your test code).

Third Party Libraries

Please do not use any third party libraries beyond the standard Python runtime libraries. For security reasons we will be unable to accept any solution that uses third-party libraries.

Pressing the ***TaskDetails*** button with *DifficultyLevel.**LEVEL_2*** set:

Polynomial Level 2

Levels of Difficulty

There are two levels of difficulty for this question (you are currently working to level-1):

- Level 1 - simplest level (basic input and calculations)
- Level 2 - medium complexity (level-1 + varied input sources)

The general scenario and rules described below (level-1 rules) apply across all two levels of difficulty for this question. Each level of difficulty adds additional rules and builds on the previous and is designed to give you the opportunity to demonstrate your design, technical and coding skills.

The level of difficulty that you decide to deliver in your final solution is entirely up to you - but remember we are looking for high quality design and coding skills.

Scenario & Rules: Level-1

In this question you'll be doing a mathematical calculation, with a coding twist. You need to evaluate multiple polynomial equations, i.e. for a given polynomial equation you'll calculate the value of y for a given value of x.

The polynomial will have the following form:

$$\mathbf{y = 3x^4 + 2x^3 - 6x^2 + 8x - 2}$$

- the number of terms in the polynomial will not exceed 10 terms
- the value of the multiplier will not exceed 10
- the value of X will not exceed 50
- all numbers used (the power, multiplier, constant and value of X) will always be positive integers

Now here is the coding twist:

The code that you write to evaluate the equation
must ONLY utilise the mathematical operators
ADD and SUBTRACT.

You **are not allowed** to use mathematical operators such as multiplication, division, modulas, "raised to the power of", logarithms or any mathematical functions/operators other than ADD or SUBTRACT. If the code you submit uses any method other than simple ADD or SUBTRACT during the calculation process the solution will be rejected.

Use of the increment/decrement operator ++ or -- (for example: i++ as a loop counter) is allowed within the code of your solution, but the value should not be used as part of an addition or subtraction within the polynomial calculation itself.

Level-2 Specifics:

The format of the input data defining the polynomial may vary in this level.

There are now two forms of input data defining the polynomial and a full test run will use BOTH forms randomly selected at approximately 50/50. **The business analysts driving the requirements for the application have also indicated that other input formats may be used in the future** - however these have not yet been specified and as yet are not included in the test data generated for this level.

The new formats that you are required to support are as follows:

Input Format 1 - json:

This is the same json format used in Level-1 except the json is prefixed with a string indicating the input type as json, thus:

```
json:{
          "xValue": 5,
          "terms":
             [
                {
                   "power": 1,
                   "multiplier": 8,
                   "action": "add"
                },
                {
                   "power": 0,
                   "multiplier": 2,
                   "action": "subtract"
                },
                {
                   "power": 4,
                   "multiplier": 3,
                   "action": "add"
                },
                {
                   "power": 3,
                   "multiplier": 2,
                   "action": "add"
                },
                {
                   "power": 2,
                   "multiplier": 6,
                   "action": "subtract"
                }
             ]
       }
```

Input Format 2 - numeric:

This is a new format and uses a highly compact numeric & mathematical-notation. This format has a prefix string indicating the new input type thus:

```
numeric:x=5;y=+8.x^1-2.x^0+3.x^4-6.x^2+2x^3

This format uses:
    a dot . to mean multiply or times
    the hat ^ to mean "to the power of"


    hence the input data above can be read as:
    x equals 5
    y equals 8 times X to the power of 1
             minus 2 times X to the power of zero
             plus 3 times X to the power of four
             minus 6 times X to the power of 2
             plus two times X to the power of three

    note:    the plus or minus sign will ALWAYS be
             present on the multiplier of the first
             term even if that multiplier is zero
```

Testing Your Solution

The Assessment Application will test your solution when you press the ***Test My Solution*** button. The application will generate lots of random input data and fire this at you solution, one test at a time (i.e. not simultaneously). In this level-2 question BOTH formats will be used at an approximate ratio of 50/50. You will see the pass/fail count increase as each input data is tested.

Once you have all tests passing AND you have coded to your chosen difficulty level AND you are satisfied with the quality of code your have written - **you are ready to submit your solution** - please see ***General Info*** for instructions on submitting your solution.

Testing Your Solution

The data representing a single polynomial is provided on each line of the input file:

LEVEL_2_<date/time>.txt

The expected answer for a given equation is provided on the following line of the file (i.e. the value of y).

Each file for Level_2 contains a set of 250 polynomials with each polynomial being randomly generated according to the criteria specified above.

Your task is to read the file content, calculate the value of y for each input string (according the rules defined above) and compare this to the expected value – ensuring you have calculated the value correctly.

Once you have all test cases passing AND you have coded to your chosen difficulty level AND you are satisfied with the quality of code you have written - **you are ready to submit your solution** - please see ***General Info*** for instructions on submitting your solution.

What We're Looking For

In this assessment exercise we are assessing your design and coding skills so your code should be of the highest quality – well designed using OOD/OOP practices, readable, reliable, robust and maintainable. Please use any techniques that you feel appropriate for such high quality code and maintainability.

If you write any supporting unit test code, please feel free to submit this to us as part of the assessment (you MAY use mathematical operators other than ADD and SUBTRACT in your test code).

Third Party Libraries

Please do not use any third party libraries beyond the standard Python runtime libraries. For security reasons we will be unable to accept any solution that uses third-party libraries.