

Problem A. Reduce a Fraction [50 points]

Problem

Create a class named *Fraction* having instance variables for numerator and denominator and a class named *IrreducibleFraction* which inherits class *Fraction*. Ensure your code follow the method descriptions described below.

Methods

- *Fraction.__init__()*: Initializer of the class *Fraction*
- *Fraction.getNumerator()*, *getDenominator()*: accessor of each local variable.
- *Fraction.setNumerator()*, *setDenominator()*: mutator of each local variable.
- *Fraction.print()*: print the *Fraction* class like example I/O below.
- *IrreducibleFraction.__init__()*: Initializer of the class *IrreducibleFraction*. Use *super()* function to initialize the class variable and modify the class variable by dividing variable by their greatest common divisor calculated by *IrreducibleFraction._GCD()* methods.
- *IrreducibleFraction._GCD()*: returns the greatest common divisor (GCD) of two nonzero integers.
- *IrreducibleFraction.print()*: print the *IrreducibleFraction* like example I/O below.

Restrictions

- Numerator and denominator should be taken as int.
- Use Skeleton Code. Both files must be in the same directory.
- Do not modify main function.

Skeleton Code

HW04_A_classes_(NAME).py

```
class Fraction():
    def __init__(self, numerator=0, denominator=1):
        ...
    def getNumerator(self):
        ...
    def setNumerator(self, value):
        ...
    def getDenominator(self):
        ...
    def setDenominator(self, value):
        ...
    def print(self):
        ...

class IrreducibleFraction(Fraction):
    def __init__(self, numerator=0, denominator=1):
        ...
    def _GCD(self, m, n):
        ...
    def print(self):
        ...
```

HW04_A_main_(NAME).py

```
from HW04_A_classes_(NAME) import *

def main():
    numerator = eval(input('Enter the Numerator: '))
    denominator = eval(input('Enter the Denominator: '))

    fraction = Fraction(numerator, denominator)
    fraction.print()

    reduced_fraction = IrreducibleFraction(numerator, denominator)
    reduced_fraction.print()

main()
```

Example I/O

```
Enter the numerator: 930
Enter the Denominator: 2170

The fraction is 930/2170
The reduced fraction is 3/7
```

Submit format

- HW04_A_classes_(NAME).py
- HW04_A_main_(NAME).py

Problem B. Rock, Scissors, Paper [50 points]

Problem

Write a program to play a three-game matches of “rock, scissors, paper” between a person and a computer. The program must use a class *Human* and *Computer* inherits from class *Contestant*. The class *Contestant* should have instance variables for *name* and *score*. A possible output is shown in the example I/O section below, where the last line should be changed to “TIE” in case of a tie.

Methods

- `Contestant.__init__()`: initializer of the class *Contestant*.
- `Contestant.getName()`, `getScore()`: accessor of each local variable.
- `Contestant.setName()`, `setScore()`: mutator of each local variable.
- `makeChoice()`: Human requests a choice from rock, scissors, paper as input (repeats the request until the input is valid) and returns the choice. Computer randomly makes the choice and returns it.

Restrictions

- Use Skeleton Code. Both files must be in the same directory.
- Do not modify `playGame(h, c)` and `judge(choiceH, choiceC)` function.
- Use random library to implement `Computer.makeChoice` method.

Skeleton Code

HW04_B_classes_(NAME).py

```
import random

class Contestant:
    def __init__(self, name="", score=0):
        ...
    def getName(self):
        ...
    def getScore(self):
        ...
    def setScore(self,value):
        ...

class Human(Contestant):
    def makeChoice(self):
        ...

class Computer(Contestant):
    def makeChoice(self):
        ...
```

HW04_B_main_(NAME).py

```
from HW04_B_classes_(NAME) import *

def main():
    ...

def playGame(h, c):
    choiceH = h.makeChoice()
    choiceC = c.makeChoice()
```

```
    if choiceH == choiceC:
        pass
    elif judge(choiceH, choiceC):
        h.setScore(h.getScore() + 1)
    else:
        c.setScore(c.getScore() + 1)

def judge(choiceH, choiceC):
    if ((choiceH == 'rock' and choiceC == 'scissors') or
        (choiceH == 'paper' and choiceC == 'rock') or
        (choiceH == 'scissors' and choiceC == 'paper')):
        return True
    else:
        return False

main()
```

Example I/O

```
Enter name of human: Garry
Enter name of computer: alphaGo

Garry, enter your choice: rock
alphaGo chooses paper
Garry: 0, alphaGo: 1

Garry, enter your choice: potato
invalid choice potato
Garry, enter your choice: scissors
alphaGo chooses paper
Garry: 1, alphaGo: 1

Garry, enter your choice: paper
alphaGo chooses rock
Garry: 2, alphaGo: 1

GARRY WIN
```

```
Enter name of human: Garry
Enter name of computer: alphaGo

Garry, enter your choice: rock
alphaGo chooses paper
Garry: 0, alphaGo: 1

Garry, enter your choice: scissors
alphaGo chooses paper
Garry: 1, alphaGo: 1

Garry, enter your choice: paper
alphaGo chooses paper
Garry: 1, alphaGo: 1

TIE
```

Submit format

- HW04_B_classes_(NAME).py
- HW04_B_main_(NAME).py