**Python Problem Set 3**

1. **PrimeCentury.py**

During the 21st century, there are many “prime” years, a year that is a prime number. For example, 2011 was a prime year because 2011 is a prime number. You are to write a program that will find all of the prime years between 2000 and 2099, inclusive. In numerical order, print the prime years for the 21st century.

For this problem, make a function called is\_prime that takes an integer parameter. It returns True if the number is prime and False otherwise. Use a loop to call the function.

**Input:**

None

**Output:**

2003

2011

2017

2027

2029

2039

2053

2063

2069

2081

2083

2087

2089

2099

1. **Pixels.py**  
    Calculate the percentage of black pixels in an image. Black pixels are represented by a B while non black pixels are represented by an N. For each image, calculate the percentage of black pixels to the nearest tenth.

**Input:**

line1 = "BBBBBBBBBB"

line2 = "BBNNBBNNBB"

line3 = "BBNNBBNNBB"

line4 = "BBBBBBBBBB"

line5 = "BBBBNNBBBB"

line6 = "BBNBBBBNBB"

line7 = "BBBNNNNBBB"

line8 = "BBBBBBBBBB"

**Output:**

80.0 %

1. **Lucky.py**

A lucky number is any positive integer whose sum of the digits are even. For example, the number 12345 is unlucky because 1+2+3+4+5=15 and 15 is an odd number. Given any number N, determine if the number is lucky or unlucky.

For this problem, make a function called sum\_digits that takes an integer parameter. It returns the sum of the digits in the number. Use a loop to call the function.

**Input:**

numbers = [58264, 24864, 125689, 121212353545]

Print Lucky if lucky or NOT Lucky if not lucky followed by the number.

**Output:**

NOT Lucky 58264

Lucky 24864

NOT Lucky 125689

Lucky 121212353545

1. **Accounts.py**  
   The school accountant needs help with her bookkeeping duties. She gets a report from the school district that lists all of the deposits made during the current month for all of the school’s clubs. She needs you to write a program that will print the total deposits for each club for the given month.

**Input:**

deposits = ["0434512", "03145234", "012341347", "0511112345", "0475746", "03654534", "02112"]

* The first two digits of the integer indicates the club that made the deposit (only 5 clubs ).
* The remaining digits indicate the deposit to the nearest penny without a decimal point

For example, 0434512 indicates that club 4 deposited 345.12.

Print the clubs in order followed by the total amount deposited.

The output does NOT need to be right aligned.

**Output:**

1 23413.47

2 1.12

3 7997.68

4 1102.58

5 111123.45

1. **Recommendation.py**  
   Mr. FavoriteTeacher was asked by Billy to write a letter of recommendation. Unfortunately, Billy just realized the deadline to submit the letter was tomorrow. So, Mr. FavoriteTeacher had one day to write the letter. Since Mr. FavoriteTeacher did not have time to write a unique letter for Billy, he decided to take a previous written letter and change the names. But, the letter he “borrowed” was for a girl, and of course, Billy is a boy. Your job is to replace all references to Sally (the girl) with Billy(the boy). Also, you must change the pronoun references from feminine to masculine.

**Input:**

recLetter = "Sally is an exceptional student. She is awesome. She is brilliant. She really knows how to program. One of Sally’s best traits is her ability to problem solve. When faced with a difficult problem, she analyzes the tasks at hand, and comes up with unique solution. Because of her character and hard work ethic, I would like to recommend Sally for everything she wants."

**Output:**

Billy is an exceptional student. He is awesome. He is brilliant. He really knows how to program. One of Billy’s best traits is his ability to problem solve. When faced with a difficult problem, he analyzes the tasks at hand, and comes up with unique solution. Because of his character and hard work ethic, I would like to recommend Billy for everything he wants.

HINT:  
To replace a section of a string with something else try

inText = text to be inserted

deText = text to be deleted

inPoint = index where text is to be inserted

recLetter = recLetter[ :inPoint ] + inText + recLetter[ inPoint + len(deletedText) : ]

1. **SuperLucky.py**

A “super” lucky number is a number that has a prime number as one of its digits. An “ultimate” lucky number is a “super” lucky number that has a prime number of prime numbers. The number 1 is NOT a prime number.

For this problem, make a function called is\_prime that takes an integer parameter. It returns True if the number is prime and False otherwise. Use a loop to call the function.

**PLEASE NOTE:** To receive full credit, you must use modulus somewhere in your solution. Otherwise, the best grade that you will receive a maximum of 90% for this problem.

For example:

1269 has only 1 prime number, the number 2 = SUPER lucky number

1234 has only 2 prime numbers, the numbers 2 and 3 = ULTIMATE lucky number

1224 has only 2 prime numbers, the numbers 2 and 2 = ULTIMATE lucky number

2357 has only 4 prime numbers, the numbers 2, 3, 5, 7 = SUPER lucky number

**Input:**

nums = [1269, 1234, 1224, 2357, 527869, 555555, 4469]

Print “SUPER” if the number is super lucky.

Print “ULTIMATE” if the number is ultimate lucky

Print “not lucky” if the number has no prime digits.

**Output:**

SUPER

ULTIMATE

ULTIMATE

SUPER

ULTIMATE

SUPER

not lucky

1. **Palindrome.py**

A palindrome is word that is spelled the same forwards and backwards. Some famous palindromes are “radar”, “Hannah”, “kayak”, “racecar”. Write a program that will take a list of words and determine if they are palindromes. When writing your program, be sure you ignore the letter case.

Write a function called reverse that takes a string parameter. It returns the string in reverse order. Use a loop to call the function.

**Input:**

words = ["radar", "banana", "alababa", "Hannah", "racecar"]

Print “PALINDROME” if the word is a palindrome, or “just a word” if it is not a palindrome.

**Output:**

PALINDROME

just a word

just a word

PALINDROME

PALINDROME

1. **Dvorak.py**  
   In the 1870’s C.L. Sholes invented what is now called the standard QWERTY keyboard. It is said that he arranged the keys to make typists type slowly enough to keep the keys on their manual typewriter from jamming.

In the 1930’s, August Dvorak invented a keyboard that placed the vowels and most common consonants on the home row so the typing rhythm would cause the hands to alternate thus increasing the typing speed and decrease the likelihood of the typewriter’s keys jamming. The home row on a Dvorak keyboard does about 70% of the work compared to about 32% on a QWERTY keyboard.

You job is to take text that was typed with the Dvorak keyboard and convert it into text that would be typed by the standard QWERTY keyboard.

QWERTY Keyboard: SPACE q w e r t y u i o p a s d f g h j k l ; z x c v b n m , . /

Dvorak Keyboard: SPACE ‘ , . p y f g c r l a o e u i d t n s - ; q j k x b m w v z

**Input:**

dvorakText = ",. jab dak. a ,rbe.pugs ycm. cb jrmlgy.p ojc.bj.v"

**Output:**

we can have a wonderful time in computer science.

1. **GradeCalculator.py**

It is the end of the six weeks and Mr. Teacher needs to average grades. Since he is a nice teacher, he will turn the lowest daily grade into a 100. Next, he will “throw out” the lowest remaining daily grade. Additionally, Mr. Teacher has given 2 tests during the grading period. Mr. Teacher uses weighted averages to calculate the overall average. Daily grades count as 40% of the overall average while major grades count for 60% of the overall average. Given a list of grades, determine the student’s average. Daily grades are designated with a “D” while major grades are designated with an “M”.

**Input:**

grades=["D 100","D 50", "D 92","D 87","M 83","D 0","D 73","M 86"]

To calculate the averages:

1. Turn the lowest daily grade into a 100: the 6th grade becomes a 100

grades=[“D 100”,“D 50”, “D 92”,“D 87”,“M 83”,“D 100”,“D 73”,“M 86”]

1. Delete the lowest daily grade from the list: the 2nd grade will be deleted  
   grades=[“D 100”,“D 92”,“D 87”,“M 83”,“D 100”,“D 73”,“M 86”]  
   50 deleted
2. Calculate the average of the daily grades and multiply by .40

100 + 92 + 87 +100 + 73 = 452,

dailyAverage = 452 / 5 = 90.4,

dailyWeightedAverage = 90.4 x .40 = 36.16

1. Calculate the average of the major grade and multiply by .60  
   83 + 86 = 169  
   majorAverage = 169 / 2 = 84.5  
   MajorWeightedAverage = 84.5 \* .60 = 50.7
2. Round the the daily grade percentage + the major grade percentage

gradeAverage = dailyWeightedAverage + majorWeightedAverage  
gradeAverage = 36.16 + 50.7 = 86.86

Print the students rounded Grade Average in the following format

**Output:**

GRADE AVERAGE: 87