ICS 31, Summer Session 10-wk, 2017

Lab 2

Problem 1

Write a function called `leapYear` which takes a year (an integer) as an argument and returns True if that year is a leap year, False if it is not a leap year. Write a program which prompts the user to type in a year and uses your `leapYear` function to determine whether or not the year is a leap year.

Example Interaction

```
Enter a year: 1997
1997 is not a leap year.

Enter a year: 1992

1992 is a leap year.
```

Problem 2

Write a function called `guessGame` which starts a guessing game. The computer picks a random number between 1 and 100 and you need to guess the number. If you guess incorrectly then the program tells you if you are too high or too low. If you guess correctly then the computer tells you how many guesses you made and the game is over. Call your `guessGame` function to demonstrate that it works.

Example Interaction

```
Guessing my number between 1 and 100.

Enter a number: 62
Too high. Try again: 32
Too low. Try again: 51
Too low. Try
again: 56

Congratulations! You got it in 4 guesses.
```

Problem 3

Write a function called `largestFactor` which takes an integer as an argument and returns the largest factor of its argument (besides itself). Write a program which prompts the user to type in an integer and uses your `largestFactor` function to determine the largest factor and print it.

Example Interaction

```
Enter a number: 25
Largest factor is 5

Enter a number: 20
Largest factor is 10
```

Problem 4

Write a function called `addDigits` which takes an integer as an argument and returns the sum of the digits in the argument. Write a program which prompts the user to enter an integer and uses your `addDigits` function to compute the sum of the digits and print it.

Example Interaction

```
Enter a number: 25
Digit sum is 7

Enter a number: 458
Digit sum is 17
```

Problem 5

Write a program which prompts the user to enter a sequence of integers and prints out the largest difference between a consecutive pair of integers in the sequence. For example, in the sequence [1, 5, 4] the differences between the first consecutive pair is -4 (1-5), and the difference between the second consecutive pair is 1 (5-4). Of these two differences, the largest difference is 1, so your program should print 1.

Example Interaction

```
Enter a sequence: 1,5,4
Largest consecutive difference is 1

Enter a sequence: 7,6,5,4,3,1
Largest consecutive difference is 2
```

Problem 6

Say you're a professor and you want to compute a student's average grade, but you want to drop the lowest and highest grades when computing the average. Write a function called `adjustedAvg` which takes a list of grades as an argument and returns the average of all of the grades while dropping the lowest and highest grades when computing the average. Each grade is an integer between 0 and 100. If

there are multiple copies of the lowest or highest grades in the list, ignore only one copy. Assume that the list contains at least 3 grades. Write a program which prompts the user to enter a comma-separated list of grades and uses your `adjustedAvg` function to compute the adjusted average and prints it out.

Example Interaction

```
Enter a sequence of grades: 20,100,70,30
Adjusted average is 50

Enter a sequence of grades: 20,80,95,20,30
Adjusted average is 43.33
```

Problem 7

Write a function called `atEnd` which takes two strings as arguments and returns True if either of the strings appear at the end of the other string. For example, if the two strings are "gohomenow" and "now", then the function should return True because "now" is at the end of "gohomenow". However, if the arguments are "home" and "gohomenow" then the function should return False since "home" is not at the end of "gohomenow", and "gohomenow" is not at the end of "home". Write a program which prompts the user to enter two strings, separated by a space, and uses your `atEnd` function to determine if one string is at the end of the other string.

Example Interaction

```
Enter two strings: gohomenow now
One string is at the end of the other string

Enter two strings: home gohomenow
One string is not at the end of the other string
```

Problem 8

Write a function called `findHarris` which takes a string as an argument and returns True if that string contains "harris" where "harris" is not directly preceded by a period, ".". For example, your function should return True for the string "gggharrisjjj" but it should return False for the string "ggg.harrisjjj". Write a program which prompts the user to enter a string and uses your `findHarris` function to determine if "harris" occurs without the preceding period.

Example Interaction

```
Enter a string: gggharrisjjj
True

Enter a string: ggg.harrisjjj
```

```
False
```

Problem 9

Write a function called `myAvg` which accepts a list of floats as an argument and returns the average value of the numbers in the list. However, each float must be rounded to the nearest integer before computing the average. For example, the computed average of the numbers 1.4,1.8,3.2 would be 2 because 1.4 is rounded to 1, 1.8 is rounded to 2, and 3.2 is rounded to 3. Do not use the `round` or the `math.ceil` functions. Write a program which prompts the user to enter a comma-separated list of numbers and uses your `myAvg` function to compute the average and prints it out.

Example Interaction

```
Enter a sequence of numbers: 1.4,1.8,3.2
Average is 2
```

Problem 10

Assume you have a recipe for a pie that needs 1 apple, 2 oranges, and 3 pickles. Write a function `pieNum` which takes three arguments, the count of apples, oranges, and pickles available, and returns the maximum number of pies that you can make with those ingredients. The first argument is the number of apples, the second argument is the number of oranges, and the third argument is the number of pickles. For example, if you make the call `pieNum(1,2,3)` should return 1 because those ingredients are enough for only 1 pie. The call `pieNum(3,100,100)` should return 3 since only 3 pies can be made with 3 apples. Write a program which prompts the user to enter the number of each ingredient and uses your `pieNum` function to compute and print the number of pies which can be made.

Example Interaction

```
How many apples?: 3
How many oranges?: 100
How many pickles?: 100

3 pies can be made
```