

Week 9 Practice Problems

#1. Write a program that creates a copy of grades.txt, by reading each grade from the original file, and then writing that value to a file of a different name.

We'll be looking at a number of different ways that this problem can be solved (see Ch 10 in your textbook). You should write four functions each solving the problem using different methods for reading the file.

#2. Open the file onetwo.txt with a simple text editor (e.g., Notepad on Windows, or TextEdit on a Mac), and you'll discover a sentence, but with each word on a separate line:

```
One,  
two,  
buckle  
my  
shoe.
```

Remember that although you can't see them, the file contains newline characters at the end of each line. Write a program that reads the input file, and writes the words to the screen, all on one line; for example:

```
One, two, buckle my shoe.
```

As with Q1, you should write a number of functions each taking in the filename and returning the one-line string. Each function should solve the problem using different methods for reading the file.

#3.

- a) Modify the functions you wrote for Q1 to use default parameters for the input filename.
- b) Modify the functions you wrote in Q2 to use a default separator character (instead of always a space) to separate the different lines in the file.

#4. This is a problem that arose a few years ago in the processing of the marks for Midterm 1 because Blackboard (the equivalent of Quercus that we were using at the time) and Crowdmark did not talk to each other perfectly.

You are given two CSV files: A Blackboard file (bb.csv) containing an entry for each student and a Crowdmark file (crowdmarks.csv) also containing a set of entries for each

APS106 - Practice Problems

2 of 2

student. Both contain the mark for Midterm #1 (see the files) but unfortunately, due to some system mis-matches:

1. Not all students in the Blackboard file are in the Crowdmark file, and
2. Some of the marks for Midterm #1 do not match.

You need to write code to read in both files and print out the id# of each student who is not in the Crowdmark file and each students whose marks do not match. For example, the beginning of your output may look like:

```
Error: Mark mismatch for 277ccd0d-0efb-40c5-a6a2-22cab7601823 41.0 2.0
Error: Mark mismatch for b8ca83f6-a4e7-41e8-97d9-85588fb4108f 43.0 9.0
Error: Mark mismatch for 48755b18-6f1b-4c4c-8ff2-420dd1be402c 34.0 38.0
Error: Mark mismatch for 67a42c7b-18cd-4762-86e6-25b6ee3cd1c7 38.0 2.0
Error: Mark mismatch for 62a597fa-ef5e-42d0-9475-843be3c6b473 33.0 32.0
Error: Mark mismatch for c2c950af-ddf4-47ef-978a-c125e96b529e 9.0 37.0
67fe5434-2eb9-41ec-b977-f508bd819f22 not in Crowdmark file.
Error: Mark mismatch for 66eb7226-7582-4f52-b366-15351abb53c6 24.0 33.0
```

The order of the students in each file may be different.

You probably want to define and implement a number of functions. Try to write clear and understandable code.

Need a hint? Post some ideas of how you think you might approach this (i.e., an Algorithm Plan) to piazza and we'll comment.

#5. Write a function to convert a positive base-10 integer to binary. Your function should take in a positive integer and return *a string of 0s and 1s* corresponding to the binary value of the value of the parameter. You may have to do a bit of research to figure out how to convert a base-10 to integer to binary.

Your program should prompt the user for a positive integer and exit if the input is not a positive integer. Then you should call your function and print the results.