

# Graduate Assessment: Good Match

## The scenario

A local tennis club has decided to organise a tournament open to the general public as a way to promote tennis and the club itself.

They have advertised widely and received many responses from people that are interested. As with any major tennis tournament there will be a mixed doubles event. Partners will be allocated according to an algorithm they have devised and you have been approached to write the program.

## The problem

Calculate the match percentage between two peoples first names.

The sentence to be processed will read as {name1} matches {name2}, for example  
**'Jack matches Jill'**

- Starting from the left, read the first character in the sentence and count how many times that character occurs.
- Move on to the next character that has not been counted yet
- Repeat until all characters in the sentence have been counted
- You should now have a number showing how many times each character occurred in the sentence

Eg.

- Jack matches Jill
- Jack matches Jill
  - 2
- Jack matches Jill
  - 22
- Jack matches Jill
  - 222
- Jack matches Jill
  - 2221
- Jack matches Jill
  - 22211
- Jack matches Jill
  - 222111

- ~~Jack matches Jill~~
  - 2221111
- ~~Jack matches Jill~~

- 22211111
- ~~Jack matches Jill~~
  - 222111111
- ~~Jack matches Jill~~
  - 2221111111
- ~~Jack matches Jill~~
  - 22211111112

This number now needs to be reduced to a 2 digit number.

- Add the left most and right most number that has not been added yet and put its sum at the end of the result.
- If there is only one number left add that number to the end of the result
- Repeat this process until there are only 2 digits left in the final string

Eg.

- 22211111112
  - 4
- 222111111+2
  - 43
- 22211111+2
  - 433
- 222+111+1+2
  - 4332
- 222+1+1+1+2
  - 43322
- 222+1+1+1+2
  - 433221
- 43322+
  - 5
- 43322+
  - 55
- 43322+
  - 555
- 555
  - 10
- 555
  - 105
- 105
  - 6
- 105
  - 60

The result is 60

The output should read 'Jack matches Jill 60%'

If the percentage is more than 80%, so for example if the output was 82% the output should read "Jack matches Jill 82%, good match"

## **The requirements**

**1. Write a program that accepts two strings and calculates the match percentage as above.**

1. The program should only accept alphabetic characters and should gracefully handle invalid input
2. Casing should not matter
3. If the result is greater than or equal to 80, 'good match' should be appended to the output string.

**2. Modify the program to accept a CSV file as input.**

1. Eg
2. The results should be listed from the highest percentage to the lowest. Ie. order by percentage descending
3. If multiple results are the same, order them alphabetically.
4. The CSV file will contain a string followed by a character indicating gender. The gender character will be either m or f
  1. Kimberly, f
  2. Jason, m
  3. Billy, m
  4. Trini, f
  5. Tommy, m
  6. Zack, m
  7. Billy, f
  8. Jason, m
5. Read the data, group them by the gender indicator, you will now have two sets of data, one with males, one with females.

6. A single set should not contain duplicates, From the example data above, Jason should only be in the male set once, however Billy will be in the male set as well as the female set.
  7. Run the good match program for every entry in the first set against every entry in the second set. Store the results
  8. Print the results in a textfile called output.txt
3. **This is optional**
1. Save logs in a separate text file - what you log is up to you
    1. Examples of logs could include
      1. Execution times

2. When input was excluded because it was badly formatted or did not meet the criteria
2. Run the good match program on the data sets in reverse and get an average good match score for each combination.
3. Create a front end that accepts two names as input and displays the result on the page