

Due September 25rd, 2013

1. Part 1 - (75 points)

Write a program in C++ called **wazoo** that reads a comma separated variable (CSV) file storing information about different animals. Each row will contain data about an animal. The first line of the file will be the header line as below:

ANIMAL TYPE, NAME, AGE, FRIENDLINESS, WEIGHT, HUNGER

After the file has been read your program should accept the following commands on standard input:

1. EXIT
 - a. The program should say the below and exit gracefully.
 - i. "THANK YOU COME AGAIN!"
2. FEED N
 - a. N (as a floating point number) is the number of food items to feed the animal
 - b. The animals should be sorted by weight from lightest to heaviest.
 - c. Their hunger factor should be decremented by one
 - d. Their weight should increase by 10% of their weight * number of food items
 - i. E.g. a 100 lb animal fed 2 food items should weigh 120 lbs after feeding.
3. PET
 - a. The animals should be sorted by friendliness from meanest to friendliest.
 - b. Their friendliness factor should be incremented by one.
4. SPEAK
 - a. The animals should be printed to standard output in their current order.
 - b. The animal data should be comma separated as shown below. This should match the input file format.
 - c. You must prefix each animal with a saying based on their animal type.

bow-wow, DOG, FIDO, 15, 9, 185, 5
meow, CAT, BOJANGLES, 1, -4, 10, 50
moo, COW, BESSY, 4, 1, 185, 5

Requirements and Notes

1. Commands should be accepted case insensitive.
2. Animal types: dog, cat, cow.
3. Animal Speeches:
 - a. Dog = bow-wow
 - b. Cat = meow
 - c. Cow = moo
4. Each data column is separated by a column in the input file.
5. The friendlier the animal, the larger the value.

6. The hungrier the animal, the larger the value.
7. Weights must be positive. The program should exit with the following message if false:
 - a. Incorrect Weight Found
8. The animal's age must be positive. The program should exit with the following message if false:
 - a. Incorrect Age Found
9. You can assume that the file will be correctly formatted (e.g. no text in numeric fields, no missing lines, no empty lines.)
10. Your program should read and store the animals in a single container, e.g. vector.
11. Your program should not display any other text, even when waiting for input.
12. You can use a vector to store the animals.

Example of executing program on command line:

commandPrompt> wazoo animals.csv

2. Part 2 (25 points)

In the same program, **wazoo**, Use template classes to implement a binary search tree (BST). The search tree should be used to store each animal. When the user types "FIND" on the standard input, the program should search for the animal by case-insensitive name:

FIND FIDO

If Fido exists, FIDO's stats should be returned along with his pecking order suffixed to the output in CSV format:

DOG, FIDO, 15, 9, 185, 5, 10

Where 10 here would be the 10th position in the feed or petting list. So if FIDO is the meanest animal, his pecking order would be 0 when the PET command issued.

If Fido does not exist, the following error message should be displayed:

"That animal does not exist in this zoo."

Duplicate Animals

All animal names should be unique. If FIDO already exists once, you should the animals name with roman numerals.

DOG, FIDO II, 15, 9, 185, 5

If there are four Fido's then the output would be as follows:

DOG, FIDO IV, 15, 9, 185, 5

Requirements and Notes

1. You may not use any STL or 3rd party tree libraries or objects. You must write your own BST.
2. You can assume that the roman numerals will go up to ten at most (e.g. X)

Grading Criteria

Inheritance

Template Classes

Correctness (includes not crashing and the output format meets the above criteria)

Submission

1. Your submission must comply with syllabus (see make directions).
2. You must submit your source code electronically per syllabus directions.
3. If the program does not compile, no credit is awarded. The homework will be tested on *elec.tricity.wsu.edu* and you should use the g++ compiler.
4. **You must turn in a hardcopy of the assignment per direction in syllabus or it will not be graded.**
5. No late submissions.

I STRONGLY RECOMMEND THAT YOU CAREFULLY DESIGN YOUR SOLUTION FIRST! WRITE DOWN ALL OF THE REQUIREMENTS IN A FORMAT THAT YOU UNDERSTAND.