

Homework 4:

With a Snap of My Fingers...

Due date: May 18 at 11:59pm.

You will not demo this homework to a lab leader; instead, you will upload it to Dropbox when you are *complete*, **even if it is after the due date**. You must also turn in your handwritten answers to the **Discussion Questions** (at the end of this document) **before the start of the final exam**. You can (and should) turn in the Discussion Questions even if you aren't finished with the code.

Overview

Billionaire superhero Tony Stark has decided to give away his fortune to half the galaxy's inhabitants. When he snaps his fingers, Tony's powerful A.I. assistant Friday will identify individuals who will each receive an equal share of Tony's total accumulated wealth. Friday is a computer, so it can't *really* pick half of all people at random... so instead, it will use a formula based on an individual's name to determine whether they gain a share of the money. You will program this formula and write an application to interact with it.

Strings and Characters

We know that a string is a sequence of zero or more characters. A computer doesn't really understand what a "character" is – it deals only with numbers, so when we see the character 'A' in our program, the computer actually considers that character to be a specific integer. Using a rule for assigning integers to each printable character called an **encoding**, a computer can then be programmed to work with strings by pretending that it's working with integers.

(You should note that uppercase and lowercase letters are different printable characters, and are given different values in an encoding.)

The most common encoding in computers is called **UTF-8**, which is derived from an older encoding called **ASCII**. Every printable (and some unprintable) English letter, digit, or symbol was given a different number with the ASCII encoding system (example: A is number 65); UTF-8 takes those same values and expands them to include millions of other symbols used in non-English languages. The next time you write a text to a friend or a URL in your browser, take a moment to appreciate that all you're really doing is recording a bunch of integers.

Java uses a related format called UTF-16 for its strings and characters. Given a **char** variable, we can convert that **char** to its integer value in UTF-16 by casting it to an **int**:

```
char a = 'X';
int value = (int)a;
```

We can then do math with **value**, effectively manipulating the integer assigned to the letter 'X' in UTF-16 (which will be the same as in the ASCII and UTF-8 encodings).

Friday's Winners

Friday will choose winners to inherit Tony's fortune based on the characters in each contestant's name. Given a name like T'Challa, Friday will convert each of the **letters** in the contestant's name to their integer value in UTF-16 encoding, **skipping** any non-letter like '. With each letter converted to a number, Friday will sum all those integer values. It will then find the **remainder** of dividing that sum by the length of the given name, this time **including** any non-letters. If the resulting remainder is **even**, then the contestant wins part of Tony's fortune! If the sum is **odd**, however, that person loses, and their hopes and dreams vanish into a cloud of dust.

For example, Friday will decide if T'Challa is a winner as such:

```
T -> 84
C -> 67
h -> 104
a -> 97
l -> 108
l -> 108
a -> 97
-----
Sum = 665
Divided by length (8) = 83 remainder 1
Not a winner :(
```

Functions

You will write the following functions in a Java program:

1. **calculateNameScore**: this function takes a **String** parameter and returns the **int** score that Friday would assign to the given name.

(a) Function behavior:

- i. Iterates through each character in the given **String** parameter, converts it to an integer if it is a letter, and sums up those integers.
- ii. Returns the sum.

(b) Restrictions:

- i. Do **not** use a **Scanner** to read input from the user.
- ii. Do **not** print.

(c) Hints:

- i. You will need to look up a way to determine if an individual character is a letter. It's quite simple.
- ii. There are two ways to iterate through the characters of a **String**:
 - A. Use a **for** loop to iterate a counter up to the length of the string, and for each value of the counter, use the **.charAt** function to retrieve the character at that position in the string:

```
for (int i = 0; i < x.length(); i++) {
    char c = x.charAt(i);
    // use c here
}
```

- B. Use an “enhanced” **for** loop that looks more like Python:

```
for (char c : x.toCharArray()) {
    // use c here
}
```

2. **isWinner**: this function takes a **String** and determines (true or false) whether that name is a winner according to Friday's formula.

(a) Function behavior:

- i. Uses **calculateNameScore** to sum the letters of the given **String** variable, then returns true if and only if the returned score is a “winner” according to Friday.

(b) Restrictions:

- i. Do **not** use a **Scanner** to read input from the user.

- ii. Do **not** print.
- 3. `getName`: this function takes no parameters; it asks the user to enter a name, validates that at least one character was entered, and returns the entered string only if it is valid.
 - (a) Function behavior:
 - i. Construct a **Scanner**.
 - ii. Ask the user to input a name.
 - iii. Use the `.nextLine()` function on your **Scanner** to read an entire line of text from the user. Save the result into a **String** variable.
 - iv. Check the `.length()` of the **String** variable. If the user entered no characters, repeat until they do.
 - v. Return the validated string.
- 4. `main`: the entry point of your program:
 - (a) Function behavior:
 - i. Get the user's name by calling `getName`.
 - ii. Check to see if the user is a winner.
 - iii. Print a congratulatory message if the user is a winner, otherwise console them.
 - A. You **must** include the user's name as part of the message you print here.
 - iv. Repeat this process until they enter the name "quit".
 - (b) Hints:
 - i. We haven't learned this yet, but you **cannot** use `==` to compare strings for equality in Java. It sucks.
 - ii. Instead, use the `.equals` function on strings, as in `x.equals("quit")`, which will be true if and only if the string variable `x` is the word "quit".

Example Output

User input is in *italics*. These do not cover every possible test case. It is up to you to thoroughly test your program before submitting it.

```
Please enter your name:
Neal Terrell
Congratulations, Neal Terrell. You win!
```

```
Please enter your name:
Peter Parker
Sorry, Peter Parker. You have to go now.
```

```
Please enter your name:
I Am Groot!
Sorry, I Am Groot!. You have to go now.
```

Discussion Questions

In addition to uploading your code to Dropbox, you must also turn in **handwritten** answers to the following questions.

1. Determine the integer output of Friday's formula for each of the following names. Show your work similar to the worked example for the name T'Challa.
 - (a) Nebula
 - (b) Bo Fu
 - (c) Donald Trump
2. Characters can be subtracted in Java and most languages, since internally they are simply integers: to subtract two characters, determine their corresponding integer encodings and subtract those. Use an ASCII code chart (Google it) or write a Java program to determine the answer to the following arithmetic operations on characters:
 - (a) '9' - '0'
 - (b) 'F' - 'A'
 - (c) 'z' - 'a'
3. Use what you learned in this assignment (and its discussion questions) to determine the output of the following Java code fragment:

```
// in a main():
String number = "482";
int a0 = number.charAt(2) - '0'; // the symbol 0 is a "zero"
int a1 = (number.charAt(1) - '0') * 10;
int a2 = (number.charAt(0) - '0') * 100;
System.out.println(a0 + a1 + a2);
```