

CS209 – Assignment1

Please write a program named “WordCount” that generate an encode table from a Chinese document. This dictionary records every Chinese character (limit in the range of 0x4e00~0x9fa5) in the file (excluding punctuations) and the corresponding encoding according the args[2](target character set), and counts the number of the character. The character, encoding and number of each character occupy one line, separated by commas. The encoding is printed to file using hexadecimal format.

Input: args [0]:file name

args [1]: character set of args[0]

args [2]: target character set

args [3]: sort key ("char", "code" or "count")

Output: A UTF-8 .txt file named args [2]+"_Dict_From_"+args [0]. Each character, corresponding encode and number in the file occupies one line separated by commas.

If args [3] is "char", data is sorted by character's value from small to large.

If args [3] is "code", data is sorted by encoding's value from small to large.

If args [3] is "count", data is sorted by count from large to small.

Here is a sample run:

For example, here is a file named "sample.txt", which charset is GB18030, target charset is UTF-8, sort key is "code".

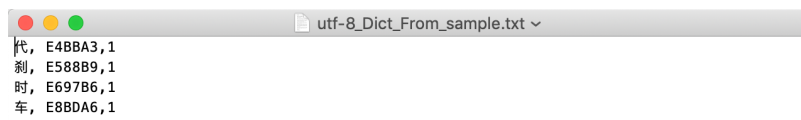


If run the command:

```
java WordCount sample.txt GB18030 UTF-8 code
```

The output file should be like this:

CS209A SPRING2019



```
代, E4BBA3, 1
刹, E588B9, 1
时, E697B6, 1
车, E8BDA6, 1
```

Here is another file:

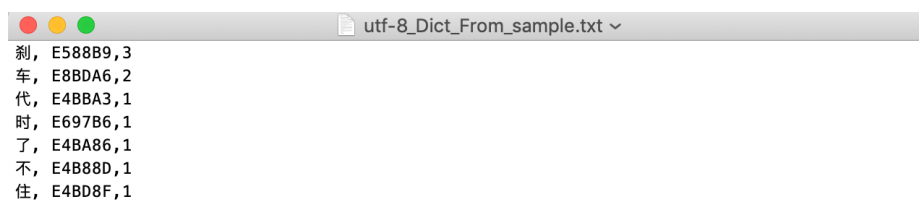


```
刹车时代刹车刹不住了
```

If run the command:

```
java WordCount sample.txt GB18030 UTF-8 count
```

The output file should be like this:



```
刹, E588B9, 3
车, E8BDA6, 2
代, E4BBA3, 1
时, E697B6, 1
了, E4BA86, 1
不, E4B88D, 1
住, E4BD8F, 1
```

VERY IMPORTANT (FOR THIS LAB AND THE NEXT ONES!)

We try in this course to have assignment submissions graded by scripts whenever possible. Those scripts use techniques frequently used in professional software development: they run a series of tests and check whether the program passes or fails each test. For every assignment, we'll try to crash your program or make it misbehave. Every failed test will mean points off. The assignment description specifies which tests will be run, so that you can check your program thoroughly before submitting it.

It's very important:

1. That you respect what is specified in the assignment, naming, input and output (no additional message, just the required result), otherwise it will be understood as a wrong result by the script.
2. That you check that your program behaves well - no Java stack dump - for all the test cases.

What you must submit (upload it to Sakai):

- The `.java` file that contains your code

What will be tested when grading your program:

1. That your program isn't the same as someone else's
2. That your program compiles correctly (javac)
3. That the result is obtained in a reasonable time