

**READ THIS FIRST:**

Do your best to do every item on your own; if you cannot immediately do an item, go on to others and then come back to it later. Please check the resources section if you have any problems and talk with your professor if there are any further questions.

**Due: Wednesday, January 24, 2018.**

**Goals:**

- Test your intallation of the Java compiler
- Test your intallation of the Java VM
- Practice getting around the command line compiling and running Java programs.
- Practice getting around in and using GitHub.
- Explain some key concepts we covered in class.
- Get some easy lab points.

---

**Description:**

In cryptography you receive a message, known as *plaintext*, and produce a encrypted message, known as *cyphertext*. One of the first steps in encryption is the conversion of symbols to numbers. Write a program that takes as input a string from ‘a’ to ‘z’ or ‘A’ to ‘Z’ and transforms it to integers from 0 to 25. If you encounter a space, it should be mapped to 26.

**Input:**

The input has  $L$  lines; each line has  $M$  words; every word has  $N$  letters. The input consists of letters in the english alphabet (a-z or A-Z) possibly separated by spaces. The input will have at least one line and 50 at the most,  $1 \leq L \leq 50$ ; a line will have at least one word and 500 at the most,  $1 \leq M \leq 500$ ; and a word will have at least one letter and 45 at the most,  $1 \leq N \leq 45$ .

**Output:**

The output is composed of  $L$  lines with numbers from 0 to 26 corresponding to each letter in every word each separated by a space.

Sample Input 1:

```
Hello
```

Sample Output 1:

```
7 4 11 11 14
```

Sample Input 2:

```
Cryptography is fun
```

Sample Output 2:

```
2 17 24 15 19 14 6 17 0 15 7 24 26 8 18 26 5 20 13
```

Sample Input 3:

```
A  
car usually  
has four tires
```

Sample Output 3:

```
0  
2 0 17 26 20 18 20 0 11 11 24  
7 0 18 26 5 14 20 17 26 19 8 17 4 18
```

### Requirements:

Create a method with the following signature `int[] str2int(String plainText)` that receives a string of plaintext corresponding to a line of input, and returns the corresponding numbers as an array of integers. Also, make sure you follow the style guidelines <http://www.reev.us/mscs630s18/style.html> that were given for this course.

**Resources:**

- Your textbook (Stanoyevitch)!
  - Project submission guidelines for this course: [www.reev.us/mscs630s18/project\\_submission.html](http://www.reev.us/mscs630s18/project_submission.html)
  - Coding style guidelines for this course: [www.reev.us/mscs630s18/style.html](http://www.reev.us/mscs630s18/style.html)
  - “How to” use the command line “shell”: [www.reev.us/mscs630s18/shell.html](http://www.reev.us/mscs630s18/shell.html)
  - The official Java reference: <http://docs.oracle.com/javase/tutorial/collections/TOC.html>
  - Stack Overflow Java Tag: <http://stackoverflow.com/questions/tagged/java>
- 

**Submission:**

- Push your work to your GitHub repository before the due date (see the top of this document). Remember to include your name, the date, and the assignment in the (copious, meaningful, and accurate) commit messages. Then **double check your files are on GitHub** and that your professor has been added as a collaborator (you hopefully did this in Homework 0), his GitHub username is **pablorp80**.
- Make sure that your program (\*.java) is in a folder called **labs/1/** inside your repository folder. Your repository folder (say **mscs630lastname**) should look something like this:

```
mscs630lastname/  
├── labs/ ... This directory will have all your labs, organized as one directory per  
│   │   │   lab, where you will have any .java files as required.  
│   ├── 1/ ... Files for this specific lab would go in here: i.e.,  
│   │   │   your .java file and test cases.  
│   ├── 2/  
│   └── ...  
├── hw/  
└── prj/
```