**Homework Turnin**

**Name:**Su Wang   
**Account:**sw1507 (sw1507@uw.edu)  
**Student ID:**1627344   
**Section:**AC   
**Course:**CSE 143 17su   
**Assignment:**a1

**Receipt ID:**c0f6fbf3f09e62d4962d3a256be99420

**Turnin Successful!**

Please **save or print a copy of this page** for your records. A receipt has also been emailed to you, but you should still save or print a copy of this web page to make certain that you have proof of your turnin. **If your files become lost or damaged and you do not have any record of your receipt, we will not be able to give you any credit for your work.**

The following file(s) were received:

**LetterInventory.java   (8114 bytes)**

1. /\*\*
2. \* This program is the homework1 for CSE 143 2017Summer
3. \* Written by: Su Wang, enrolled in quiz session AC, Student ID: 1627344
4. \* LetterInventory is a list integers, each stores the value of the number of a
5. **\* letter's appearance in a given string. Each value represents a letter in**
6. \* alphabetic sequence.
7. \*/
8. **public** **class** LetterInventory {
9. **private** **int**[] elementData;
10. **private int size;**
11. **public** **static** **final** **int** INVENTORY\_CAPACITY = 26;
13. /\*\*
14. \* Constructs an initial LetterInventory with capacity of 26 to represent
15. **\* each letter in alphabet with each counts equals zero;**
16. \*/
17. **public** LetterInventory() {
18. **this**("");
19. }
21. /\*\*
22. \* The constructor takes a string of data as the parameter to construct an
23. \* initial LetterInventory with each number represents the counts of letter
24. \* in data given.
25. **\***
26. \* @param data
27. \* can be a string of data, uppercase and lowercase will be
28. \* treated as same, non-alphabetic characters will be ignored.
29. \*/
30. **public LetterInventory(String data) {**
31. elementData = **new** **int**[INVENTORY\_CAPACITY];
32. size = 0;
33. **int** index;
34. **for** (**int** i = 0; i < data.length(); i++) {
35. **int charToInteger = (char) data.charAt(i);**
36. // to eliminate the non-alphabetic characters.
37. **if** ((charToInteger <= 'z' & charToInteger >= 'a') || (charToInteger <= 'Z' & charToInteger >= 'A')) {
38. index = getIndex(data.charAt(i));
39. elementData[index]++;
40. **size++;**
41. }
42. }
43. }
45. **/\*\***
46. \* Gets the size of the inventory.
47. \*
48. \* @return the sum of all the counts in the inventory.
49. \*/
50. **public int size() {**
51. **return** size;
52. }
54. /\*\*
55. **\* Examines if the list is empty or not.**
56. \*
57. \* @return True if the inventory is empty, return False if not.
58. \*/
59. **public** **boolean** isEmpty() {
60. **if (size() == 0) {**
61. **return** **true**;
62. }
63. **return** **false**;
64. }
66. /\*\*
67. \* Gets the number of times the given letter appears in the inventory.
68. \*
69. \* @param letter
70. **\* should be an alphabetic character, can be either lowercase of**
71. \* uppercase, throws IllegalArgumentException if violated.
72. \* @returns an integer, which is the number of times the given letter
73. \* appears in the inventory.
74. \*/
75. **public int get(char letter) {**
76. **int** index = getIndex(letter);
77. **return** elementData[index];
78. }
80. **/\*\***
81. \* Returns a String representation of the inventory with all the letters in
82. \* lowercases, in sorted order, and surrounded by square brackets The
83. \* returned string is in lower case.
84. \*/
85. **public String toString() {**
86. **String** allLetters = "";
87. **for** (**int** i = 0; i < elementData.length; i++) {
88. **if** (elementData[i] != 0) {
89. **for** (**int** j = 1; j <= elementData[i]; j++) {
90. **allLetters += (char) (i + 'a');// prints out the string in**
91. // lower cases
92. }
93. }
94. }
95. **return ("[" + allLetters + "]");**
96. }
98. /\*\*
99. \* Sets the count for the given letter to the given value.
100. **\***
101. \* @param letter
102. \* should be an alphabetic letter. Lowercase or in uppercase will
103. \* be treated as same, wil throw IllegalArgumentException if a
104. \* non-alphabetic character is passed.
105. **\* @param value**
106. \* should be an integer greater than or equal to zero, will throw
107. \* IllegalArgumentException if violated.
108. \*/
109. **public** **void** set(**char** letter, **int** value) {
111. **int** indexOfList = getIndex(letter);
112. **if** (elementData[indexOfList] > value) {
113. size -= elementData[indexOfList] - value;// size change equals the
114. // difference of the
115. **// change in value.**
116. } **else** **if** (elementData[indexOfList] < value) {
117. size += value - elementData[indexOfList];
118. }
119. elementData[indexOfList] = value;
120. **}**
122. /\*\*
123. \* Constructs and returns a new LetterInventory object that represents the
124. \* sum of LetterInventory and the other given LetterInventory.
125. **\* Returns a**
126. \* LetterInventory with each count equals to the sum of the two respective
127. \* counts in the two LetterInventory. The two LetterInventory being added
128. \* will not be changed by this method.
129. \*
130. **\* @param other**
131. \* should be a LetterInventory
132. \* @returns a new LetterInventory with each counts represents the sum of the
133. \* respective two counts in the LetterInventory and the given
134. \* inventory.
135. **\*/**
136. **public** LetterInventory add(LetterInventory other) {
137. LetterInventory list = **new** LetterInventory();
138. **for** (**int** i = 0; i < INVENTORY\_CAPACITY; i++) {
139. list.elementData[i] = **this**.elementData[i] + other.elementData[i];// add
140. **// up**
141. }
142. **return** list;
143. }
145. **/\*\***
146. \* Constructs and returns a new LetterInventory object that represents the
147. \* difference of LetterInventory and the other given LetterInventory. The
148. \* two Inventories being subtracted will not be changed via this method.
149. \*
150. **\* @param other**
151. \* should be a LetterInventory
152. \* @returns new LetterInventory with each count equals to the result after
153. \* subtract the count in the given Inventory from the respective
154. \* count in the LetterInventory returns null if any single count
155. **\* result is a negative number.**
156. \*/
157. **public** LetterInventory subtract(LetterInventory other) {
158. LetterInventory subtractResult = **new** LetterInventory();
159. **for** (**int** i = 0; i < INVENTORY\_CAPACITY; i++) {
160. **subtractResult.elementData[i] = this.elementData[i] - other.elementData[i];**
161. **if** (subtractResult.elementData[i] < 0) {
162. **return** **null**;
163. }
164. }
165. **return subtractResult;**
166. }
168. /\*\*
169. \* Calculates the percentage of the number of appearance for the given
170. **\* letter in this inventory.**
171. \*
172. \* @param letter
173. \* should be a alphabetic character in either lower case or upper
174. \* case. Throws IllegalArgumentException if non-alphabetic
175. **\* character is passed.**
176. \* @returns a decimal number with one decimal point, which represents the
177. \* percentage of the given letter appeared in the LetterInventory.
178. \*/
179. **public** **double** getLetterPercentage(**char** letter) {
180. **double size = size();**
181. **double** numbersOfLetter = get(letter);
182. **double** percentage = numbersOfLetter / size;
183. **return** percentage;
184. }
186. /\*\*
187. \* Gets the index of the item (which is the number of times the given letter
188. \* appears) in the LetterInventory List. the index value is calculated as
189. \* the numeric order of the given letter in the alphabetic list. For
190. **\* example, if you pass a you will get 1; if you pass z you will get 26.**
191. \*
192. \* @param letter
193. \* should be an alphabetic character, in either lower or upper
194. \* case, throws IllegalArgumentException if violated.
195. **\* @returns an integer, which is the index number in the LetterInventory**
196. \* list the given letter represents
197. \*/
198. **private** **int** getIndex(**char** letter) {
199. **int** letterToInteger = letter;
200. **int index;**
201. **if** (letterToInteger <= 'z' & letterToInteger >= 'a') {// lower case
202. index = letterToInteger - 'a';
203. } **else** **if** (letterToInteger <= 'Z' & letterToInteger >= 'A') {// upper
204. // case
205. **index = letterToInteger - 'A';**
206. } **else** {
207. **throw** **new** **IllegalArgumentException**();
208. }
209. **return** index;
210. **}**
211. }

**decodedCryptogram.txt   (8540 bytes)**

a light in the attic by shel silverstein --------------------------------- somebody has to somebody has to go polish the stars they're looking a little bit dull somebody has to go polish the stars for the eagles and starlings and gulls have all been complaining they're tarnished and worn they say they want new ones we cannot afford so please get your rags and your polishing jars somebody has to go polish the stars --------------------------------- something missing i remember i put on my socks i remember i put on my shoes i remember i put on my tie that was printed in beautiful purples and blues i remember i put on my coat to look perfectly grand at the dance yet i feel there is something i may have forgot what is it what is it --------------------------------- rock and roll band if we were a rock and roll band we'd travel all over the land we'd play and we'd sing and wear spangly things if we were a rock and roll band if we were a rock and roll band and we were up there on the stand the people would hear us and love us and cheer us hurray for that rock and roll band if we were a rock and roll band then we'd have a million fans we'd goggle and laugh and sign autographs if we were a rock and roll band if we were a rock and roll band the people would all kiss our hands we'd be millionaires and have extra long hair if we were a rock and roll band but we aint no rock and roll band were just seven kids in the sand with homemade guitars and pails and jars and drums of potato chip cans just seven kids in the sand talking and wavin our hands and dreamin and thinking oh wouldn't it be grand if we were a rock and roll band --------------------------------- monsters i've met i met a ghost but he didn't want my head he only wanted to know the way to denver i met a devil but he didn't want my soul he only wanted to borrow my bike awhile i met a vampire but he didn't want my blood he only wanted two nickles for a dime i keep meeting all the right people at all the wrong times --------------------------------- put something in draw a crazy picture write a nutty poem sing a mumble-gumble song whistle through your comb do a loony-goony dance cross the kitchen floor put something silly in the world that aint been there before --------------------------------- picture puzzle piece one picture puzzle piece lyin on the sidewalk one picture puzzle piece soakin in the rain it might be a button of blue on the coat of the woman who lived in a shoe it might be a magical bean or a fold in the red velvet robe of a queen it might be the one little bite of the apple her stepmother gave to snow white it might be the veil of a bride or a bottle with some evil genie inside it might be a small tuft of hair on the big bouncy belly of bobo the bear it might be a bit of the cloak of the witch of the west as she melted to smoke it might be a shadowy trace of a tear that runs down an angels face nothing has more possibilities than one old wet picture puzzle piece --------------------------------- signals when the light is green you go when the light is red you stop but what do you do when the light is blue with orange and lavender spots --------------------------------- shaking geraldine now stop shaking that cow for heavens sake for your sake and the cows sake thats the dumbest way i've seen to make a milk shake --------------------------------- what did what did the carrot say to the wheat lettuce rest i'm feeling beet what did the paper say to the pen i feel quite all write my friend what did the teapot say to the chalk nothing you silly teapots can't talk --------------------------------- prayer of the selfish child now i lay me down to sleep i pray the lord my soul to keep and if i die before i wake i pray the lord my toys to break so none of the other kids can use them amen --------------------------------- the sitter mrs mctwitter the baby-sitter i think shes a little bit crazy she thinks baby-sitters supposeed to sit upon the baby --------------------------------- stop thief policemen policemen help me please someone went and stole my knees id chase him down but i suspect my feet and legs just won't connect --------------------------------- how not to have to dry the dishes if you have to dry the dishes such an awful boring chore if you have to dry the dishes stead of going to the store if you have to dry the dishes and you drop one on the floor maybe they won't let you dry the dishes anymore --------------------------------- hammock grandma sent the hammock the good lord sent the breeze im here to do the swinging now whos gonna move the trees --------------------------------- moon catching net ive made me a moon catching net and i'm going hunting tonight ill run along swinging it over my head and grab for that big ball of light so tommorow just look at the sky and if there's no moon you can bet ive found what i sought and i finally caught the moon in my moon catching net but if the moons still shining there look close underneath and you'll get a clear look at me in the sky swinging free with a star in my moon catching net --------------------------------- how many how much how many slams in an old screen door depends how loud you shut it how many slices in a bread depends how thin you cut it how much good inside a day depends how good you live them how much love inside a friend depends how much you give them --------------------------------- a light in the attic there's a light on in the attic though the house is dark and shuttered i can see a flickerin flutter and i know what its about there's a light on in the attic i can see it from the outside and i know you're on the inside looking out --------------------------------- the giving tree by shel silverstein once there was a tree and she loved a little boy and everyday the boy would come and he would gather her leaves and make them into crowns and play king of the forest he would climb up her trunk and swing from her branches and eat apples and they would play hide and go seek and when he was tired he would sleep in her shade and the boy loved the tree very much and the tree was happy but time went by and the boy grew older and the tree was often alone then one day the boy came to the tree and the tree said come boy come and climb up my trunk and swing from my branches and eat apples and play in my shade and be happy i am too big to climb and play said the boy i want to buy things and have fun i want some money? i'm sorry said the tree but i have no money i have only leaves and apples take my apples boy and sell them in the city then you will have money and you will be happy and so the boy climbed up the tree and gathered her apples and carried them away and the tree was happy but the boy stayed away for a long time and the tree was sad and then one day the boy came back and the tree shook with joy and she said come boy climb up my trunk and swing from my branches and be happy i am too busy to climb trees said the boy i want a house to keep me warm he said i want a wife and i want children and so i need a house can you give me a house i have no house said the tree the forest is my house but you may cut off my branches and build a house then you will be happy and so the boy cut off her branches and carried them away to build his house and the tree was happy but the boy stayed away for a long time and when he came back the tree was so happy she could hardly speak come boy she whispered come and play i am too old and sad to play said the boy i want a boat that will take me far away from here can you give me a boat? cut down my trunk and make a boat said the tree then you can sail away and be happy and so the boy cut down her trunk and made a boat and sailed away and the tree was happy but not really and after a long time the boy came back again i am sorry boy said the tree but i have nothing left to give you my apples are gone my teeth are too weak for apples said the boy my branches are gone said the tree you cannot swing on them i am too old to swing on branches said the boy my trunk is gone said the tree you cannot climb i am too tired to climb said the boy i am sorry sighed the tree i wish that i could give you something but i have nothing left i am just an old stump i am sorry i don't need very much now said the boy just a quiet place to sit and rest i am very tired well said the tree straightening herself up as much as she could well an old stump is good for sitting and resting come boy sit down sit down and rest and the boy did and the tree was happy