Zachary Chang

UID: 805289445

Discussion 1G

Lecture 1

Programming Assignment 5 Report

1. One of the most major issues I had to resolve was finding the number of bees in the trial word. Meeting all the criteria for being qualified to be a bee is difficult: it must not be a flower, and it must not already be in another pairing. This was a hard problem to solve without the use of arrays, so I wrote out my algorithm consisting of two arrays on a piece of paper and then translated this into actual code. I eventually decided that this algorithm using two arrays was the easiest way to track whether or not individual characters had already been paired. In addition, another obstacle was finding out an efficient way to thoroughly test my program. Instead of running my program and actually having to figure out the mystery word myself, I defined the random word rather than allowing the random word picker to choose for me. This way I was able to systematically check whether the bee and flower counters were accurate.
2. (below)

***MAIN METHOD***

*fill a 2-D array with a c-strings*

*check that at least one c-string is in the array*

*ask the user for how many rounds to play and check that the input is positive*

*repeatedly, for the number of rounds the user wants to play:*

*choose a random c-string for the mystery word*

*display round number and how long the mystery word is*

*let the user play, returning their score*

*report how many tries it took to find the mystery word*

*report the average, max, and min scores across all rounds played so far*

***PLAYONEROUND METHOD***

*check that the mystery word lies within the array of c-strings*

*repeatedly, while the mystery word has not been found:*

*allow user to input a trial word*

*check that trial word is at least 4 and at most 6 characters, report error if not*

*check that trial word is comprised only of lowercase letters, report error otherwise*

*check that trial word exists in array of c-strings, report error otherwise*

*increase score by 1*

*find the number of flowers*

*if notified that mystery word has been solved:*

*return score*

*otherwise:*

*find the number of bees and report the number of flowers and bees*

***FLOWER METHOD***

*find the length of the mystery word*

*increment flower count every time two letters match in the same position*

*if all characters are flowers:*

*signal to playOneRound that the user found the mystery word*

*return the number of flowers otherwise*

***BEE METHOD***

*fill two arrays with zeroes to begin tracking paired characters*

*find the length of the mystery word*

*find length of the trial word*

*find the flowers in the trial and mystery words and mark “flower characters” with a 1*

*repeatedly through all characters in the mystery word:*

*repeatedly through all characters in trial word:*

*move on if character is paired*

*if two characters are equal and are each not paired:*

*mark them with 1’s and increment the bee count*

*return the number of bees*

***GETMYSTERYLENGTH METHOD***

*repeatedly, through the length of the mystery word:*

*iterate until the ‘\0’ character is reached, and the index is the length of the c-string*

*return the length*