

Project #3

Due Dates: Monday, October 19 at 11:59pm

Submit: eLearning

Late Policy: -10 points per hour late

Instructions: This is an individual assignment. All work should be your own.

Objectives:

- Create a hash table.
- Use the hash table to solve a word puzzle.

Description:

The word puzzle is described in section 1.1 in the textbook and discussed further at the end of chapter 5.

A grid consisting of letters is to be checked against a dictionary of words to see if the grid contains any of the words.

The user can input a value for the rows and columns of the grid and the program will create a grid of random characters. You can limit the grid size to 20x20.

The program will read in a dictionary file (provided) and use an algorithm to solve the word puzzle.

The hash table should be a linear probing hash table of your own creation, not copied other sources. You should design and code it yourself. You can use code from the QuadraticProbingHashTable given in the text if you wish.

The program should use the algorithm described as the "second" algorithm in 1.1, which checks each word in the grid for presence in the dictionary.

The user should also have the option of using the following enhancement:
When reading the input file of words, store each prefix of the word as well. For example, if the word is "apple", store "a", "ap", "app", "appl", "apple". In the algorithm, if a prefix is not found, the rest of this string can be treated as "not found". For example, if the string is "apbum", and after checking and finding "a" and "ap" I find that "abp" is not in my dictionary, then there is no point in checking further in this direction. Note you will need to indicate for each entry whether it is a word or only a prefix of a word.

Have the program output the elapsed time in both cases.

Submit to eLearning:

WordPuzzle.java
MyHashTable.java