Anagrams

Group Project by:

- Fatima Abubakar
- Efemena Onovre
- Tamunokorite Victor Briggs
- David Usiere
- Muhammad Abdullahi Jibril

Table of Contents

Introduction

An anagram is a word or a phrase that is configured to form another word or phrase, using all the letters of the subject. The original word is known as the subject. Any word or phrase that can reconstruct the letters of another word to form a coherent English word is an anagram.

Aim Of Project

The aim of our project is to create a working and perfectly efficient program that inputs a word and outputs it’s anagrams.

Program Objectives

- The program will prompt for a file with the list of words.
- The program will return all the anagrams for the words read.

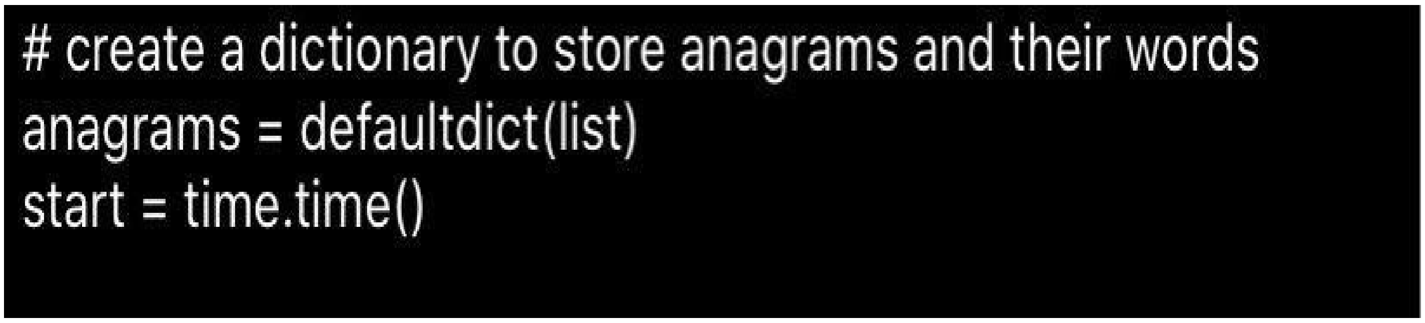
- The program will output the list of words that form the most anagrams,
- The project will output the largest anagram.

Programming Concepts Used

- Defined Functions
- Looping
- Conditions
- Len
- Count
- New Line Character (\n)
- Max
- Sort
- Lists
- Round
- Python Dictionary
- Module
- Import
- From

New Concepts Used

Dictionary: it is an unordered collection of data values which is changeable and indexed. It holds a pair of values, Key and the Key:Value. Values in the dictionary can be of any data type while keys cannot be repeated and are immutable. You access the elements of a dictionary using their keys. The use of dictionary in our code:

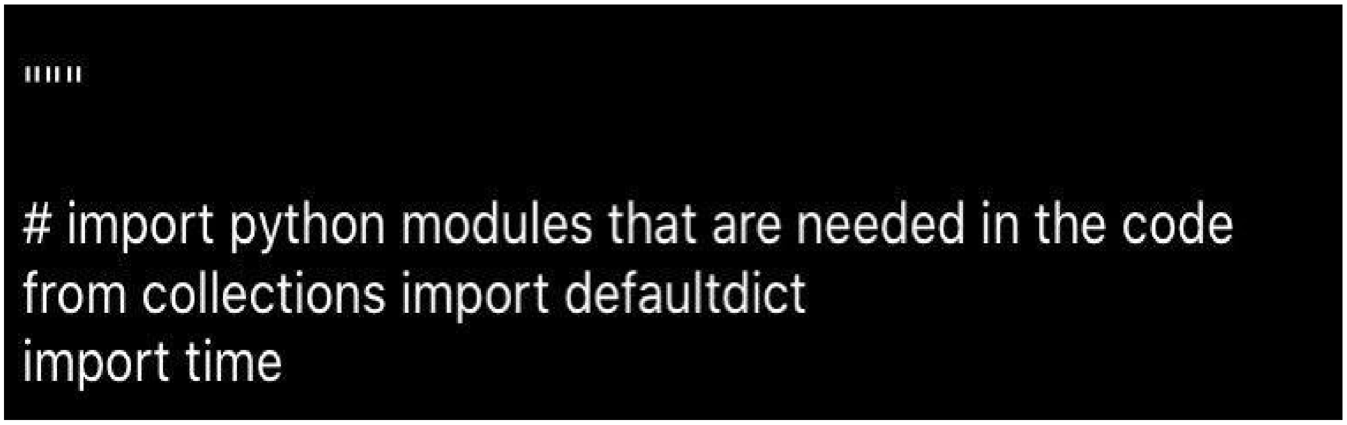


```
# create a dictionary to store anagrams and their words
anagrams = defaultdict(list)
start = time.time()
```

Module: it is a file containing a set of functions you want to use in your program, stored in another file. You create a module by saving a file with the ".py" extension.

Import: This is a statement used to bring in the content of a module to a program.

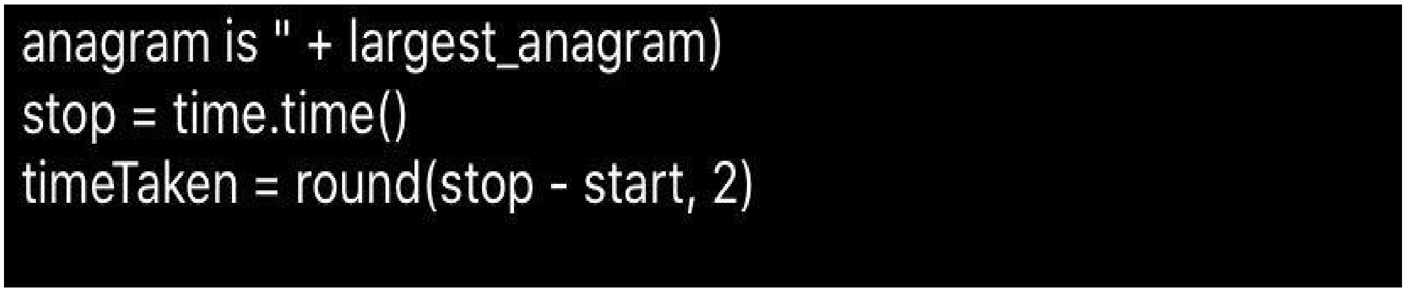
From: This is used when you want to import only parts from a Module.
The use of module, from and import in our code:



```
"""

# import python modules that are needed in the code
from collections import defaultdict
import time
```

Round: it is a function that returns a floated number, a float being a number with a decimal point, that is the rounded up version of the specified number, with a specified number of decimals. The use of round in our code:



```
anagram is " + largest_anagram)
stop = time.time()
timeTaken = round(stop - start, 2)
```

Description

The program firstly imports the *defaultdict*and *time* from the inbuilt python module collections to use later on in the code, after which it defines a function “is_anagram” and conditions it to check whether two words are anagrams of each other by returning true or false.

The program then prompts the user to enter the file name and file extension, storing the content of the file in a list and reads each word on a line with the use of the newline character (\n). A dictionary is created to store anagrams and their words, the word list is then looped through for a word to check for all the anagrams of that word and adds it to the anagram list.

The function *largest_anagram* is defined to check the anagram dictionary and find the word with the most anagram (largest anagram)..

The program outputs the list of the largest anagrams by stringing the list of the largest anagrams and also outputs the largest anagram.

Lastly the program outputs the time taken to find the list of the largest anagrams and the largest anagram by the use of the *start* and *stop* variables which contain the time started and the time ended respectively. *Start* and *stop* are subtracted and rounded to find the time taken.

FLOWCHART

The program code as well as the report have been submitted together with the flowchart of the program. Note that the flowchart was done with Microsoft word and as such, the pdf form is not accurate. Therefore, both versions of the flowchart were submitted- The Microsoft word version being the accurate version and the pdf version, to have an immutable version of the flowchart even though it is not accurate.

Conclusion

The program runs successfully as the aim was carried out and it meets all the objectives of the project.

Difficulties And Obstacles Of Project

- Execution: we had a clear vision of the work we needed to do but as inexperienced programmers execution was hard.
- Conversion of documents: With the flowchart, the pdf did not recognise some symbols used on Microsoft word and even Microsoft word did not properly execute their symbols (the flowchart loop symbol) and so proved difficult.
- Time constraint: one of our biggest obstacles was time, we wanted to do a stellar job which requires meticulousness and also get the work done in time.
- Prioritization: this is one of many projects assigned to us, getting group members to commit to the project was a painful task.
- Unfamiliarity: Our group members were not acquainted so breaking the barrier and putting heads together took quite some time and effort.

Group Members:

The group was divided into two parts, programmers and report writers in order to utilize time and ensure the success of the project and the group:

- Tamunokorite Victor Briggs: the group leader and also the programmer
- David Usiere: a programmer and also made the flowchart.
- Efemena Hilda Onovre: a programmer that made the pseudo code
- Fatima Abubakar: Report writer that made the pseudo code.
- Muhammad Abdullahi Jibril: Report writer

Reference

- WIKIPEDIA [ONLINE] https://en.m.wikipedia.org/wiki/Anagram
- W3SCHOOLS [ONLINE] https://www.w3schools.com/python/python_modules.asp https://www.w3schools.com/python/python_dictionaries.asp
- GEEKSFORGEEKS [ONLINE] https://www.google.com/amp/s/www.geeksforgeeks.org/python-dictionary/amp/