Programming Assignment Unit 8

Dept. of Computer Science, University of the People

CS 1101-01: Programming Fundamentals

Misbah Ullah

March 24, 2024

**Python code:**

# Function to read dictionary items from a file

def read\_dictionary(filename):

with open(filename, 'r') as file:

dictionary = eval(file.read())

return dictionary

# Function to invert a dictionary

def invert\_dictionary(original\_dict):

inverted\_dict = {}

for key, value in original\_dict.items():

if isinstance(value, list):

for item in value:

inverted\_dict.setdefault(item, []).append(key)

else:

inverted\_dict.setdefault(value, []).append(key)

return inverted\_dict

# Function to write the inverted dictionary to a file

def write\_dictionary(inverted\_dict, filename):

with open(filename, 'w') as file:

file.write(str(inverted\_dict))

# Input and output file paths

input\_file = 'original\_dictionary.txt'

output\_file = 'inverted\_dictionary.txt'

# Reading original dictionary from input file

original\_dict = read\_dictionary(input\_file)

# Inverting the dictionary

inverted\_dict = invert\_dictionary(original\_dict)

# Writing inverted dictionary to an output file

write\_dictionary(inverted\_dict, output\_file)

**Explanation:**

The program first defines three functions: **read\_dictionary**, **invert\_dictionary**, and **write\_dictionary**. These functions read dictionary items from a file, invert the dictionary, and write the inverted dictionary to another file.

In the **read\_dictionary** function, the file is opened in read mode, and the **eval() function** is used to convert the file contents (which should contain a valid Python dictionary) into a dictionary object.

The **invert\_dictionary** function iterates through the original dictionary, swapping keys and values. If a value is a list, it iterates through it and adds each item as a key with the original key appended to its value list in the inverted dictionary.

Finally, the **write\_dictionary** function writes the inverted dictionary to a file by converting it to a string using **str()**.

The program specifies the input and output file paths, reads the input dictionary, inverts it, and writes the output dictionary.