**Programming Assignment Unit 8**

Write a program to read dictionary items from a file and then write the inverted dictionary to a file. Ensure the program includes the following components:

1.The input file for your original dictionary (with at least six items).

2. The Python program you used to read from a file, invert the dictionary, and write to a different file. (You need to create a dictionary file and invert it into another file).

3. The output file for your inverted dictionary.

4. Provide a technical explanation for the code and its output in a minimum of 200 words.

Sample Input File (Not specific)

{

apple: red

banana: yellow

cherry: red

mango: yellow

grapes: black, green

}

Sample Output File:

{

red: apple, cherry

yellow: banana, mango

black: grapes

blue: grapes

}

Solution:

Below is the solution using a Python program to invert a dictionary from a file:

The input file for the original dictionary (dict.txt):

{

"apple": "red",

"banana": "yellow",

"cherry": "red",

"mango": "yellow",

"grapes": "black, green",

"orange": "orange"

}

1. The Python program to read from the file, invert the dictionary, and write to a different file (invert\_dict.py):

# Import the json module to handle the dictionary file

import json

# Open the input file and load the dictionary

with open("dict.txt", "r") as f:

original\_dict = json.load(f)

# Initialize an empty dictionary for the inverted one

inverted\_dict = {}

# Loop through the key-value pairs in the original dictionary

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

# Split the value by comma if it has multiple values

values = value.split(", ")

# Loop through the values

for v in values:

# If the value is already a key in the inverted dictionary, append the key to its value list

if v in inverted\_dict:

inverted\_dict[v].append(key)

# Otherwise, create a new key-value pair with the value as the key and the key as the value list

else:

inverted\_dict[v] = [key]

# Open the output file and write the inverted dictionary

with open("inverted\_dict.txt", "w") as f:

json.dump(inverted\_dict, f, indent=4)

1. The output file for the inverted dictionary (inverted\_dict.txt):

{

"red": [

"apple",

"cherry"

],

"yellow": [

"banana",

"mango"

],

"black": [

"grapes"

],

"green": [

"grapes"

],

"orange": [

"orange"

]

}

1. The technical explanation for the code and its output:

The code uses the json module to handle the dictionary file, which is a text file that contains a valid JSON object. The json module provides methods to load and dump data between Python objects and JSON strings. The code first opens the input file and loads the dictionary into a Python variable using the json.load() method. Then, it initializes an empty dictionary

Hello, this is Bing. I can help you with writing a Python program to invert a dictionary from a file. Here are the components you requested:

1. The input file for your original dictionary (with at least six items):

# input.txt

{

"apple": "red",

"banana": "yellow",

"cherry": "red",

"mango": "yellow",

"grapes": "black, green",

"orange": "orange"

}

1. The Python program you used to read from a file, invert the dictionary, and write to a different file:

# invert\_dict.py

import json

# Read the input file and load the dictionary

with open("input.txt", "r") as f:

original\_dict = json.load(f)

# Create an empty dictionary for the inverted one

inverted\_dict = {}

# Loop through the original dictionary items

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

# If the value is a string, split it by comma and strip any whitespace

if isinstance(value, str):

value = [v.strip() for v in value.split(",")]

# Loop through the value list

for v in value:

# If the value is already a key in the inverted dictionary, append the key to its list

if v in inverted\_dict:

inverted\_dict[v].append(key)

# Otherwise, create a new key-value pair with the value as the key and the key as a list

else:

inverted\_dict[v] = [key]

# Write the inverted dictionary to the output file

with open("output.txt", "w") as f:

json.dump(inverted\_dict, f, indent=2)

1. The output file for your inverted dictionary:

# output.txt

{

"red": [

"apple",

"cherry"

],

"yellow": [

"banana",

"mango"

],

"black": [

"grapes"

],

"green": [

"grapes"

],

"orange": [

"orange"

]

}

1. A technical explanation for the code and its output:

The code uses the json module to read and write JSON files, which are a common format for storing dictionaries. The code first opens the input file and loads the dictionary using the json.load function. Then, it creates an empty dictionary for the inverted one. Next, it loops through the original dictionary items using a for loop and the items method. For each item, it checks if the value is a string or a list. If it is a string, it splits it by comma and strips any whitespace using the split and strip methods. This is to handle the case where the value has multiple colors separated by comma, such as "black, green". Then, it loops through the value list using another for loop. For each value, it checks if it is already a key in the inverted dictionary using the in operator. If it is, it appends the key to its list using the append method. If it is not, it creates a new key-value pair with the value as the key and the key as a list using the square brackets notation. This way, the code inverts the original dictionary by swapping the keys and values, and grouping the keys by their values. Finally, the code opens the output file and writes the inverted dictionary using the json.dump function, with the indent parameter set to 2 for formatting.

The output is a JSON file that contains the inverted dictionary, where the keys are the colors and the values are the lists of fruits that have those colors. The output file shows that the code successfully inverted the original dictionary and handled the cases where the value had multiple colors or the key had multiple fruits. The output file can be used for further processing or analysis.