UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

1. Dictionary Concatenation

Aim

To concatenate two or more dictionaries as required.

Algorithm

Input: Any set of dictionaries
Output: A single concatenated dictionary

Step 1: Start
Step 2: Read the given dictionaries as input from the user.
Step 3: Create a new empty dictionary.
Step 4: Define a function to add the contents of the pre-existing dictionary to the new dictionary. The function accepts one of the given dictionaries and the new dictionary to which the contents should be added as parameters.
Step 3: The function runs through every key-value pair in the dictionary by accessing them through the items() method and adds them as items into the new dictionary. The function returns the new dictionary.
Step 4: In the main function, the function is called with the empty dictionary and the first dictionary to be concatenated. The new dictionary now has the contents of the first given dictionary.
Step 5: The function is again called with the new dictionary and the second dictionary as parameters. The new dictionary now has the contents of the first and the second given dictionaries.
Step 6: Call the function repeatedly until the contents of all the given dictionaries have been added to the new dictionary.
Step 7: Add a user-input key to the new dictionary and set the value as None.

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Step 8: Print the new dictionary as the concatenated dictionary.
Step 9: End

Program

```
# Python program to concatenate dictionaries.

# function to concatenate two dictionaries
def create(dic,dict):
  for k,v in dict.items():
    dic[k]=v
  return dic

#calling function for concatenation
create(dic,dic1)
create(dic,dic2)
create(dic,dic3)

#to display the key-value pairs of the dictionary
print("New:",dic)

#to display only the values in the dictionary
print("Values:",list(dic.values()))

#to display only the keys in the dictionary
print("Keys:",list(dic.keys()))

#to append one more key
x=int(input("Enter new element: "))
dic[x]=None

#to display the final dictionary
print("Final:",dic)
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Output

```
New: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
Values: [10, 20, 30, 40, 50, 60]
Keys: [1, 2, 3, 4, 5, 6]
Enter new element: 7
Final: {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60, 7:
None}
```

Results / Inferences

Program for concatenating dictionaries is written and executed.

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

2. Squares of a Range of Numbers

Aim

To store the squares of a given range of numbers in a dictionary and to print them.

Algorithm

Input: A range of numbers
Output: The squares of the range of numbers

Pre-condition: The limits must be integers.

Step 1: Start
Step 2: Define a function to return the squares of a given range of numbers. The function accepts the lower and upper limits of the range as arguments.
Step 3: The function creates an empty dictionary. For every number in the range of numbers, it finds the square by using the exponential operator and adds the number and square as a key-value pair to the dictionary.
Step 4: The function returns the dictionary.
Step 5: In the main function, the required lower limit and upper limit are obtained as input from the user.
Step 6: Call the function to return the squares with the limits as parameters and print the return value.
Step 7: End

Program

```
# Python program to print and store the squares of the numbers in a
given range in a dictionary.
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

```
# function to return the squares for the range
def square(a,b):
  d={}
  for i in range(a,b+1):
    d[i]=i**2
  return d

#user-input range
l=int(input("Lower limit: "))
u=int(input("Upper limit: "))

print(square(l,u))
```

Output

Input:

Range of 2 to 5, both inclusive

Output:

```
Lower limit: 2
Upper limit: 5
{2: 4, 3: 9, 4: 16, 5: 25}
```

Results / Inferences

Program for printing and storing the squares of a given range of numbers in a dictionary is written and executed.

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

3. Counting Positive and Negative numbers

Aim

To count the number of positive and negative numbers in a given list of numbers.

Algorithm

Input: A list of numbers
Output: The number of positive and negative numbers in the list

Pre-condition: The list must contain numeric values only.

Step 1: Start
Step 2: Define a function to count the number of positive and negative numbers in a list. The function accepts the list as the argument.
Step 3: The function reads a dictionary with two keys- positive and negative. The values for both keys are initialised to zero.
Step 4: The function runs through every element in the list and if the number is greater than 0, adds the value of the positive key by 1 and if the number is lesser than 0, adds the value of the negative key by 1.
Step 5: The function returns the dictionary.
Step 6: In the main function, read the list of numbers as input from the user.
Step 7: Call the function to find the number of positive and negative numbers with the list as the parameter and print the return value.
Step 8: End

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Program

```
# Python program to encrypt a message using the Caesar cypher.

#Function to count the positive and negative numbers
def count(l):
  d={"pos":0,"neg":0}
  for i in l:
    if i>0:
      d["pos"]+=1
    elif i<0:
      d["neg"]+=1
  return d

#to get the list as input from the user
n=int(input("Enter size of list: "))
l=[int(input("Enter num: ")) for i in range(n)]
print("List:",l)

#to print the count
print(count(l))
```

Output

Input:

```
[1, -2, -3, 4, 5]
```

Output:

```
Enter size of list: 5
Enter num: 1
Enter num: -2
Enter num: -3
Enter num: 4
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

```
Enter num: 5
List: [1, -2, -3, 4, 5]
{'pos': 3, 'neg': 2}
```

Results / Inferences

Program for counting the number of positive and negative entries present in a given list is written and executed.

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

4. Biggest Value

Aim

To find the key in a dictionary that corresponds to the entry with the largest number of values associated with it.

Algorithm

Input: A dictionary
Output: The key with the largest number of values

Step 1: Start
Step 2: Read the given dictionary.
Step 3: Define a function to find the key with the largest number of values. The function accepts the dictionary as the argument.
Step 4: The function reads a variable to hold the maximum value and initialises it to 0. The function runs through every item in the dictionary and counts the number of elements the value contains. If it is more than the maximum value, the maximum value is changed to this number of elements. The key corresponding to this value in the dictionary is stored in another variable.
Step 5: After checking every item in the dictionary, the function returns the variable holding the key corresponding to the maximum value.
Step 6: In the main program, call the function with the given dictionary as the parameter and print the return value.
Step 7: End

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Program

```
# Python program to find the key with the largest number of values in
the dictionary.

#the dictionary
animals={'L':['Lion'],'D':['Donkey','Deer'],'E':['Elephant']}

#function that returns the key with the largest number of values
def biggest(d):
  max=0
  for k,v in d.items():
    if len(v)>max:
      max=len(v)
      b=k
  return b

print("The key with the largest number of values:",biggest(animals))
```

Output

Input:

```
animals={'L':['Lion'],'D':['Donkey','Deer'],'E':['Elephant']}
```

Output:

```
The key with the largest number of values: D
```

Results / Inferences

Program for finding the key with the largest number of values is written and executed.

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

5. French-English Dictionary

Aim

To find all the French translations of a given English word by finding all the keys in a dictionary that map to a specific value.

Algorithm

Input: A list of English words
Output: The French translation of the given English words

Step 1: Start
Step 2: Read the dictionary that maps French words to English words.
Step 3: Define a function to look up all the keys corresponding to a given value. The function accepts the dictionary and the value as the arguments.
Step 4: The function reads an empty list. For every item in the dictionary, if the value in the key-value pair is equal to the required value, the function appends the key to the list.
Step 5: After checking the entire dictionary, the function returns the list of keys.
Step 6: In the main program, read the list of English words.
Step 7: Run a loop to access every word in the list.
Step 8: For each word, call the function with the given dictionary and the word as the parameters and print the return value each time.
Step 9: End

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Program

```
# Python program to find all the keys in a dictionary that map to a
specific value, i.e., all French words that map to the English word

#the dictionary
FrEn={'le':'the','la':'the','livre':'book','pomme':'apple'}

#function to find the French translation
def reverseLookup(d,val):
  l=[]
  for k,v in d.items():
    if v==val:
      l.append(k)
  return l

#the required English words
word=['the','apple','food']

for i in word:
  print("The French word for",i,"is",reverseLookup(FrEn,i))
```

Output

Input:

```
word=['the','apple','food']
```

Output:

```
The French word for the is ['le', 'la']
The French word for apple is ['pomme']
The French word for food is []
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Results / Inferences

Program for looking up the French translation for the given English words is written and executed.

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

6. Anagrams

Aim

To check if two given phrases are anagrams or not.

Algorithm

Input: Any two phrases
Output: The conclusion of the phrases being anagrams or not

Step 1: Start
Step 2: Define a function to find the frequency of each letter in a string. The function accepts the string as the argument.
Step 3: The frequency function does the following:

- Create an empty dictionary.
- Convert the string to lower case.
- In a loop, check each character in the string and proceed if the character is a letter.
- Add every letter as a key in the dictionary and increment the count by 1 for every occurrence of the letter in the string.
- Print the dictionary to show the frequency of each letter.
- Return the dictionary.

Step 4: Define a function to check if two phrases are anagrams or not. The function accepts the two phrases as parameters.
Step 5: The anagrams function does the following:

- Call the frequency function for each phrase with the phrase as the parameter.
- If the frequencies (return value) for both the phrases are equal, then print that the phrases are anagrams.
- If not, then print that they are not anagrams.

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Step 6: In the main program, read any two phrases as input from the user.
Step 7: Pass the phrases as the parameters and call the anagrams function.
Step 8: End

Program

```
# Python program to check if two given phrases are anagrams or not.

#to count the frequency of each letter in the phrase
def freq(s):
  f={}
  s=s.lower()
  for i in s:
    if i.isalpha():
      f[i]=0
  for i in s:
    if i.isalpha():
      f[i]+=1
  print(s,f)
  return f

#to check if anagrams or not
def anagram(s1,s2):
  if freq(s1)==freq(s2):
    print("Anagrams")
  else:
    print("Not anagrams")

s1=input("Enter phrase 1: ")
s2=input("Enter phrase 2: ")
anagram(s1,s2)
```

UGE1197
Programming in Python Lab
AY: 2020-2021

Krithika Swaminathan
Date: 06/03/2021
S03018

Ex. No.: 10

Output

```
Enter phrase 1: William Shakespeare
Enter phrase 2: I am a weakish speller
william shakespeare {'w': 1, 'i': 2, 'l': 2, 'a': 3, 'm':
1, 's': 2, 'h': 1, 'k': 1, 'e': 3, 'p': 1, 'r': 1}
i am a weakish speller {'i': 2, 'a': 3, 'm': 1, 'w': 1,
'e': 3, 'k': 1, 's': 2, 'h': 1, 'p': 1, 'l': 2, 'r': 1}
Anagrams

Enter phrase 1: nice to meet you
Enter phrase 2: pleased to meet you
nice to meet you {'n': 1, 'i': 1, 'c': 1, 'e': 3, 't': 2,
'o': 2, 'm': 1, 'y': 1, 'u': 1}
pleased to meet you {'p': 1, 'l': 1, 'e': 4, 'a': 1, 's':
1, 'd': 1, 't': 2, 'o': 2, 'm': 1, 'y': 1, 'u': 1}
Not anagrams
```

Results / Inferences

Program for checking if two given phrases are anagrams or not is written and executed.