***BE 1600***

***Introduction to***

***Programming and Computation***

***Python***

**Assignment 08**

40 points

**Due 12/06/2023 (11:45 A.M.)**

Assignment Objectives:

* To use Python dictionaries to store associative data
* To implement algorithms to compute elementary statistics
* To use sets for storing and fast accessing non-duplicate elements.

*Solution for this assignment will not be posted on Canvas; however, the solution of any of the assignment problems can be discussed in the class upon request of a student.*

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

1. For non-programming questions, use a word file to type your answers. Don’t use the text box on the Canvas to answer the questions or to write comments, we will not read it. State your answer clearly.
2. For programming questions, include only the source file of each programming problem.
3. Submit your files to the Canvas. You must submit your files on time; otherwise, you will receive zero.
4. Use “Add Another File” feature on Canvas to upload each additional file; do not upload the files as a compressed folder.
5. You can upload your files as many times as you like. Only the last attempt counts because the last files you uploaded are the only files your instructor will see.
6. There will be several modules on the Canvas. You need to upload your files using the correct module on the Canvas.
7. Name each file: *Assignment (assignment number)* for the word file [e.g. Assignment 02] and *Assignment (assignment number) \_ (Question number)* for each programming question [e.g. Assignment 02\_Q03].
8. To upload your file(s):

* In Course Navigation, click the ASSIGNMENTS module.
* Click the title of the assignment.
* Click the **Submit** Assignment button.
* Add **File**. ...
* Add Another **File**. ...
* **Submit** Assignment. ...
* View **Submission**.

*It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.*

***Make sure you review the Cheating & Plagiarism policy on Canvas.***

Write a program for questions Q.1. to Q.7. Submit 7 text files to Canvas by the due date.

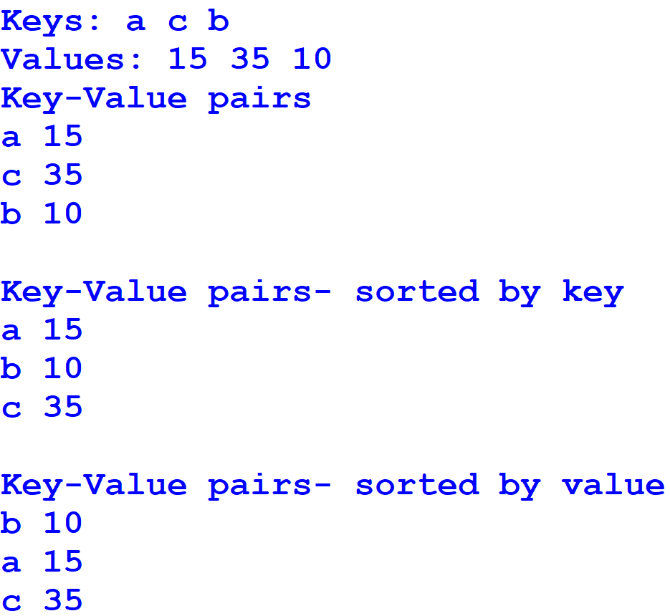
**Question 1 (5 points)**

Write a program that creates a dictionary with the following items: {'a':15 , 'c':35, 'b':10}.

Write one or more statements to:

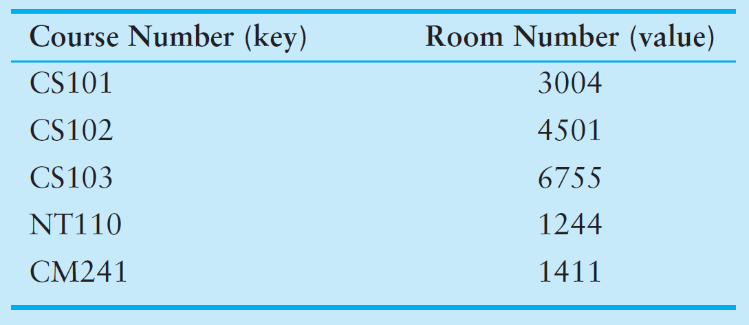
1. print all the keys.
2. print all the values.
3. print all the keys and values pairs in tabular format.
4. print all the keys and values pairs in tabular format ordered by key.
5. print all the keys and values pairs in tabular format ordered by value.

Here is a sample run:

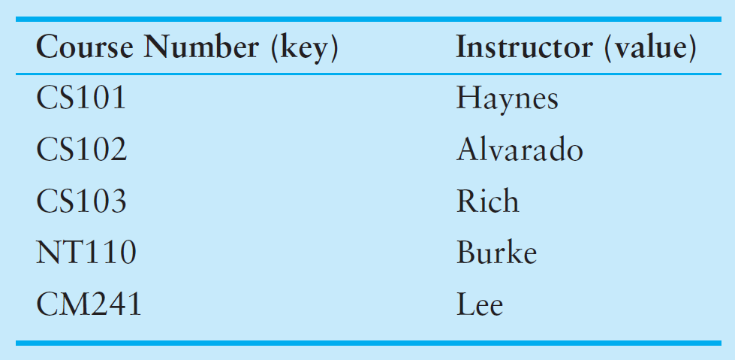


**Question 2 (6 points)**

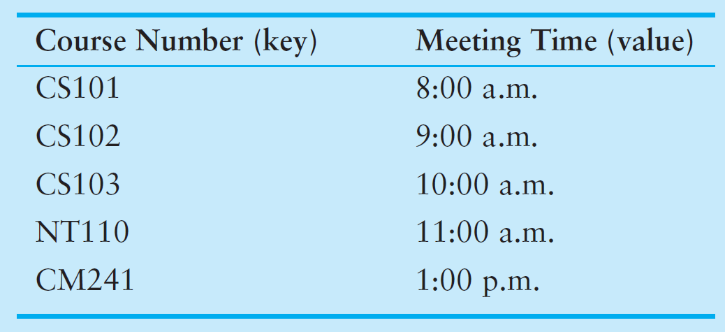
Write a program that creates a dictionary containing course numbers and the room numbers where the courses meet. The dictionary should have the following key-value pairs:



The program should also create a dictionary containing course numbers and the names of the instructors that teach each course. The dictionary should have the following key-value pairs:

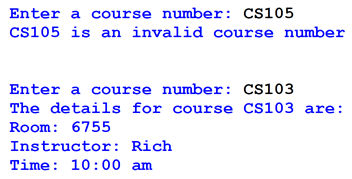


The program should also create a dictionary containing course numbers and the meeting times of each course. The dictionary should have the following key-value pairs:



The program should let the user enter a course number, and then it should display the course’s room number, instructor, and meeting time; otherwise, it should display invalid course number.

Here are sample runs:



**Question 3 (5 points)**

Create a main **function** that prompts for an integer and prints out the integer using words. For example: 138 will print “one three eight.” Use a dictionary to represent digits as keys and words as values.

Here is a sample run:

**Enter an integer to convert to words: 24**

**two four**

**Enter an integer to convert to words: 2022**

**two zero two two**

**Question 4 (6 points)**

Write a main **function** that creates a dictionary that maps countries to their capitals. Start with an empty dictionary. Ask the user to input number of countries’ names and their capitals and add them to a dictionary. E.g., capitals = { 'Argentina':'Buenos Aires', 'France':'Paris', 'US': 'Washington D.C.'}.

Prints the dictionary in a tabular format sorted by country as shown below.

Here is a sample run:  
**Enter a country and a capital comma separated (Q to quit): Japan,Tokyo**

**Enter a country and a capital comma separated (Q to quit): France,Paris**

**Enter a country and a capital comma separated (Q to quit): United States,Washington D.C.**

**Enter a country and a capital comma separated (Q to quit): Spain,Madrid**

**Enter a country and a capital comma separated (Q to quit): Q**

**COUNTRY CAPITAL**

**France Paris**

**Japan Tokyo**

**Spain Madrid**

**United States Washington D.C.**

**Question 5 (6 points)**

Write a program that have the following functions:

* A main **function** that read a person first and last name from keyboard. The main function calls the below functions.
* A **function** that takes first and last names and returns a set of the unique common letters in the first and last names (intersection).
* A **function** that takes first and last names and returns a set of all unique letters in the first and last names (union).
* A **function** that takes first and last names and returns a set of all unique letters in first name but not in last name and all unique letters in last name but not in first name (symmetric difference).

Here is a sample run:

**Enter first name: Ernest**

**Enter last name: Hemingway**

**Intersection: {'e', 'n'}**

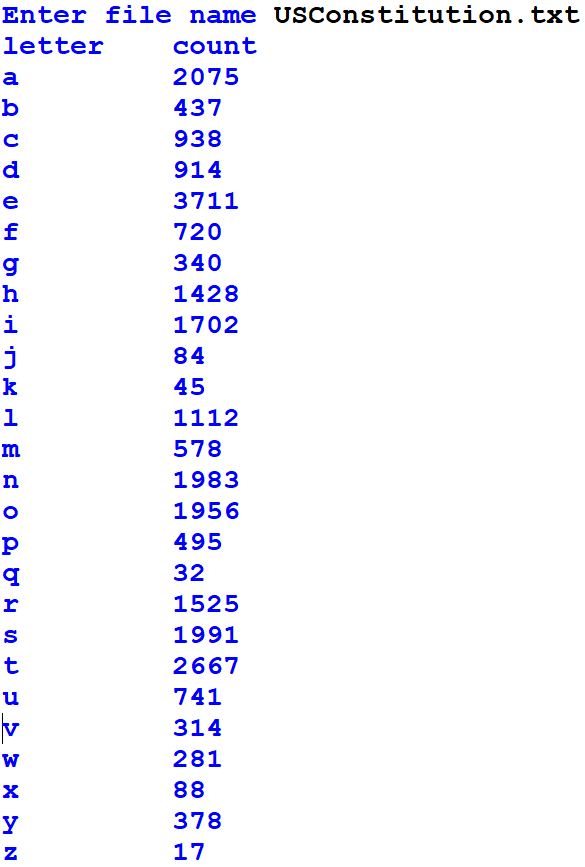
**Union: {'m', 'i', 't', 'e', 'a', 'r', 'E', 'H', 'g', 'w', 'n', 's', 'y'}**

**Symmetric: {'m', 'i', 'a', 'y', 't', 'r', 'E', 'H', 'g', 's', 'w'}**

**Question 6 (6 points)**

Write a ***main function*** that prompts for a file name and counts the frequency of each letter (*a* through *z*) using a dictionary. Make each letter lowercased and ignore punctuation. Print a histogram of the word counts . Use string format method to format the table as shown below. You may use USConstitution.txt file, included with this assignment, to test your program.

Here is a sample run:

****

**Question 7 (6 points)**

Write a program that uses a dictionary to assign “codes” to each letter of the alphabet. For example:

codes = {'A':')','a':'0','B':'(','b':'9','C':'\*','c':'8', 'D':'&','d':'7','E':'^','e':'6','F':'%','f':'5', 'G':'$','g':'4','H':'#', 'h':'3','I':'@','i':'2', 'J':'!','j':'1','K':'Z','k':'z','L':'Y','l':'y', 'M':'X','m':'x','N':'W', 'n':'w', 'O':'V','o':'v', 'P':'U', 'p':'u','Q':'T','q':'t','R':'S','r':'s', 'S':'R','s':'r','T':'Q','t':'q','U':'P','u':'p', 'V':'O','v':'o','W':'N','w':'n','X':'M', 'x':'m', 'Y':'L','y':'l','Z':'K','z':'k','!':'J','1':'j', '@':'I','2':'i','#':'H', '3':'h','$':'G','4':'g', '%':'F','5':'f','^':'E','6':'e', '&':'D','7':'d', '\*':'C','8':'c','(':'B','9':'b',')':'A','0':'a', ':':',',',':':','?':'.','.':'?','<':'>','>':'<', "'":'"','"':"'", '+':'-','-':'+','=':';',';':'=', '{':'[','[':'{','}':']',']':'}'}{ 'A' : '%', 'a' : '9', 'B' : '@', 'b' : '#', etc...}

Using this dictionary, the letter ‘A’ would be assigned the symbol ‘)’, the letter ‘a’ would be assigned the number ‘0’, the letter ‘B’ would be assigned the symbol ‘(‘, and so forth. Create a **function** that opens a specified text file, reads its contents, and then uses the dictionary to write an encrypted version of the file’s contents to a second file. Each character in the second file should contain the code for the corresponding character in the first file. Create a second **function** that opens an encrypted file and displays its decrypted contents on the screen.Write a main function that tests the above two functions.

Here is a sample run for input text file, FirstAmendment.txt

Enter the name of the input text file: FirstAmendment.txt

Enter the name of the output file to save encrypted text: Encrypt.txt

Enter the name of the encrypted input file: Encrypt.txt

|  |  |
| --- | --- |
| **FirstAmendment.txt**  The First Amendment provides several rights protections: to express ideas through speech and the press, to assemble or gather with a group to protest or for other reasons, and to ask the government to fix problems. It also protects the right to religious beliefs and practices. It prevents the government from creating or favoring a religion. | **Encrypt.txt**  Q36 %2srq )x6w7x6wq usvo276r r6o6s0y s243qr usvq68q2vwr, qv 6mus6rr 2760r q3svp43 ru6683 0w7 q36 us6rr: qv 0rr6x9y6 vs 40q36s n2q3 0 4svpu qv usvq6rq vs 5vs vq36s s60rvwr: 0w7 qv 0rz q36 4vo6swx6wq qv 52m usv9y6xr? @q 0yrv usvq68qr q36 s243q qv s6y242vpr 96y265r 0w7 us08q286r? @q us6o6wqr q36 4vo6swx6wq 5svx8s60q2w4 vs 50ovs2w4 0 s6y242vw? |

**Shell output**

The First Amendment provides several rights protections: to express ideas through speech and the press, to assemble or gather with a group to protest or for other reasons, and to ask the government to fix problems. It also protects the right to religious beliefs and practices. It prevents the government from creating or favoring a religion.