National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Object Oriented Programming**	**Course Code:**	**CS-1002**
Program:	**BS (Software Engineering)**	**Semester:**	**Spring 2023**
Deadline:	**27-Feb-2023**	**Total Marks:**	**25*4 = 100**
Section:	**BSE-2C**	**CLO**	**1**
	Assignment-1		

Instruction/Notes:

Please read the following instructions carefully before submitting the assignment.

1. Save your file with your Roll Number.
2. There should be only one cpp file.
3. It should be clear that your assignment will not get any credit if:
   - The assignment is submitted after the due date.
   - Assignment is copied (partial or full) from any source (websites, forums, students, etc.)

Important Note:

1. There shouldn't be any memory leakage in your program.
2. Make separate functions for input and output. Your main should be a sequence of function calls only.
3. You are not allowed to use global variables and goto instruction.
4. All allocations of 1D/2D pointers should be dynamic.
5. Delete the array when it is no longer needed.
6. All the data will be given by user.
7. Pass the pointers to functions instead of subscript notation [].
8. Make proper functions to solve the problems.
9. Debug your code to find errors/bugs.
10. Be confident enough that you can do it.

Task 1: [Tokenization]

Prompt the user to enter the data and get the input using character array (input should be at-lease 4 to 5 sentences long where each sentence ended with dot. You can declare a 1D character array (dynamic or static up to you, preferably dynamic) large enough to store 2000 to 3000 characters). It is better to read the data from a text file to save the input time. Your task is to Tokenize the words (here blank space should be treated as a token) entered by user and store these words in another 2D dynamic character array "data dictionary". Make sure there should not be any duplication in data dictionary. Number of rows in data dictionary should be equal to the unique words entered by the user and number of columns in each row should be one greater than the word length to store null terminator.

Example: 1 **Input:** I love Pakistan. **Output:** I Love Pakistan	**Example#2 Input:** This assignment is a piece of cake. **Output:** This assignment is a piece of cake

Task 2: [Synonyms]

Now you need to create a 3D character array "synonyms" equal to the size of unique words in data dictionary and initialize each index of "synonyms" with null pointer. Your program should iterate for each word of "data dictionary" and ask the user whether he/she wants to store synonyms of that word. If user is interested to store the synonym then ask for the count of synonyms i.e., how many synonyms he/she wants to store against that word. Get the count of synonyms from the user and allocate the required memory at that index of "3D synonyms array". Now iterate to get the input for the synonyms. To get the input for synonyms you can use a "temp 1D static character array". Once you get the synonym from user in a "temp" character array you need to allocate the memory at that particular index of "3D synonym array" equal to the size (strlen+1) of "temp" character array then copy the data character by character and then prompt the user to enter the next synonym if any. You need to repeat this task for each word in "data dictionary".

Example:1

Input Word: love
Synonym: affection

Example:2

Input Word: assignment
Synonym: assessment

Task 3: [Words in data dictionary]

Now create another 1D dynamic character array "useCase1" sufficiently large enough to store two to three thousand characters. Prompt the user to enter the data (at-least 3-4 sentences), your program should search each word of this "useCase1" in "data dictionary" and keep the count of words that are matched. You also need to print those words on console that are matched.

Bonus: Include the unmatched words of "useCase1" in "data dictionary".

Example:1

Suppose the string "I love Pakistan" is stored in data dictionary.
Input: They love Pakistan.

Output:
There are 2 words found in data dictionary. The words are as follows:
love
Pakistan

Example:2

Suppose "This assignment is piece of cake" is stored in data dictionary.
Input: This assignment is due by tomorrow.

Output:
There are 3 words found in data dictionary. The words are as follows:
This
assignment
is

Task 4: [Synonym in our stored synonym]

Now you need to replace the words in “useCase1” with their synonyms. First you need to check whether the words in “useCase1” present in “data dictionary”. If it is present then you need to check whether there exists any synonym of that word. If “yes” then replace the word of “useCase1” with its synonym.

Example: 1

Input: This is my assignment

Words	**Synonyms**
assignment	assessment

Output: This is my assessment.