# תרגיל 3 עץ Trie עבור השלמת מלים.

**מטרת התרגיל:**

כאשר המשתמש מקיש התחלת מחרוזת של תגובה עליה הוא רוצה להגיב- המערכת תחפש תגובה ההמתחילה בתת-המחרוזת אותה הוא הקיש, ותציע לו השלמות רלוונטיות.

לצורך כך, נשתמש במבנה נתונים מסוג Trie, כפי שנלמד בהרצאה.

**פירוט התרגיל – הצעה למימוש:**

הגדירו מחלקה מסוג TrieNode שתכיל:

* מערך מצביעים לבנים של הקודקוד.
* שדה בוליאני המסמן סוף מילה.
* מצביע לקודקוד האבא.

מחלקה זו תהיה מחלקה פנימית בתוך מחלקת trie.

לאחר מכן נגדיר מחלקה Trie לשמירת מאגר מחרוזות, שתממש את העץ עצמו, אשר תכיל שדה מסוג מצביע ל - TrieNode, ובנוסף את הפונקציות הבאות:

(אפשרות למימוש השורש: הקדקוד של העץ יצביע על האבא שהוא null וכך נדע כי הוא השורש).

1. הוספת מחרוזת חדשה למאגר (הפונקציה תקבל מחרוזת ותוסיף אותו לעץ).

תזכורת: הכנסה – מקרים שונים:

המילה כבר קיימת / המסלול קיים / חלק מהמסלול קיים / המסלול כולו לא קיים.

1. מחיקת מחרוזת קיימת מהמאגר (הפונקציה תקבל מחרוזת ותמחק אותה מהעץ על פי האפשרויות השונות). מחזירה FALSE במקרה שהמילה לא היתה בעץ.

תזכורת: הוצאה – מקרים שונים:

המילה לא קיימת / המילה קיימת ומסתיימת בקדקוד פנימי / המילה קיימת ומסתיימת בקדקוד פנימי / המילה קיימת ומסתיימת בקדקוד עלה / מחיקת המסלול כלפי מעלה עד הקדקוד הראשון שהוא לא עלה או שהוא מילה בפני עצמה.

1. חיפוש האם מחרוזת קיימת במאגר (פונקציה תקבל מחרוזת לחיפוש).
2. קבלת מצביע לקודקוד במאגר והחזרת כל המחרוזות השמורות בתת העץ שמתחיל מקודקוד זה (כלומר, כל המחרוזות שמתחילות מאותו קודקוד), כרשימה מילה אחר מילה, בסדר לקסיקוגרפי. שימו לב – על התכנית הראשית להדפיס את כל המילים המתחילות בתחילית שנקלטה, כולל התחילית עצמה.

ניתן להגדיר פונקציות נוספות וכן שדות נוספים בכל אחת מהמחלקות לעזר, לפי הצורך.

לשם הפשטות נניח כי הא"ב בתרגיל זה הם רק תווים של אותיות קטנות באנגלית, a-z.

השתמשו בפונקציה הראשית הבאה לצורך ביצוע כל הפעולות הנ"ל:

#include<iostream>

using namespace std;

#include <list>

#include <string>

#include "trie.h"

int main()

{

char ch;

string wrd;

Trie tr;

int comp;

cout << "\nTRIE PROGRAM" << endl;

cout << "\nChoose one of the following" << endl;

cout << "1: add a new item" << endl;

cout << "2: delete an item" << endl;

cout << "3: check if an item exists " << endl;

cout << "4: complete the possible item name " << endl;

cout << "5: exit " << endl;

do

{

cin >> ch;

switch (ch)

{

case '1':

cout << "Enter a word to insert " << endl;

cin >> wrd;

tr.insert(wrd);

break;

case '2':

cout << "Enter a word to del " << endl;

cin >> wrd;

if (!tr.del(wrd)) cout << "ERROR\n";

break;

case '3':

cout << "Enter a word to search " << endl;

cin >> wrd;

if (tr.search(wrd))

cout << "exists\n";

else

cout << "does not exist\n";

break;

case '4':

cout << "Enter a prefix to complete " << endl;

cin >> wrd;

comp = tr.printAutoSuggestions(wrd);

if (comp == 0)

cout << "No string exist with this prefix\n";

break;

case '5':cout << "bye " << endl; break;

default: cout << "ERROR " << endl; break;

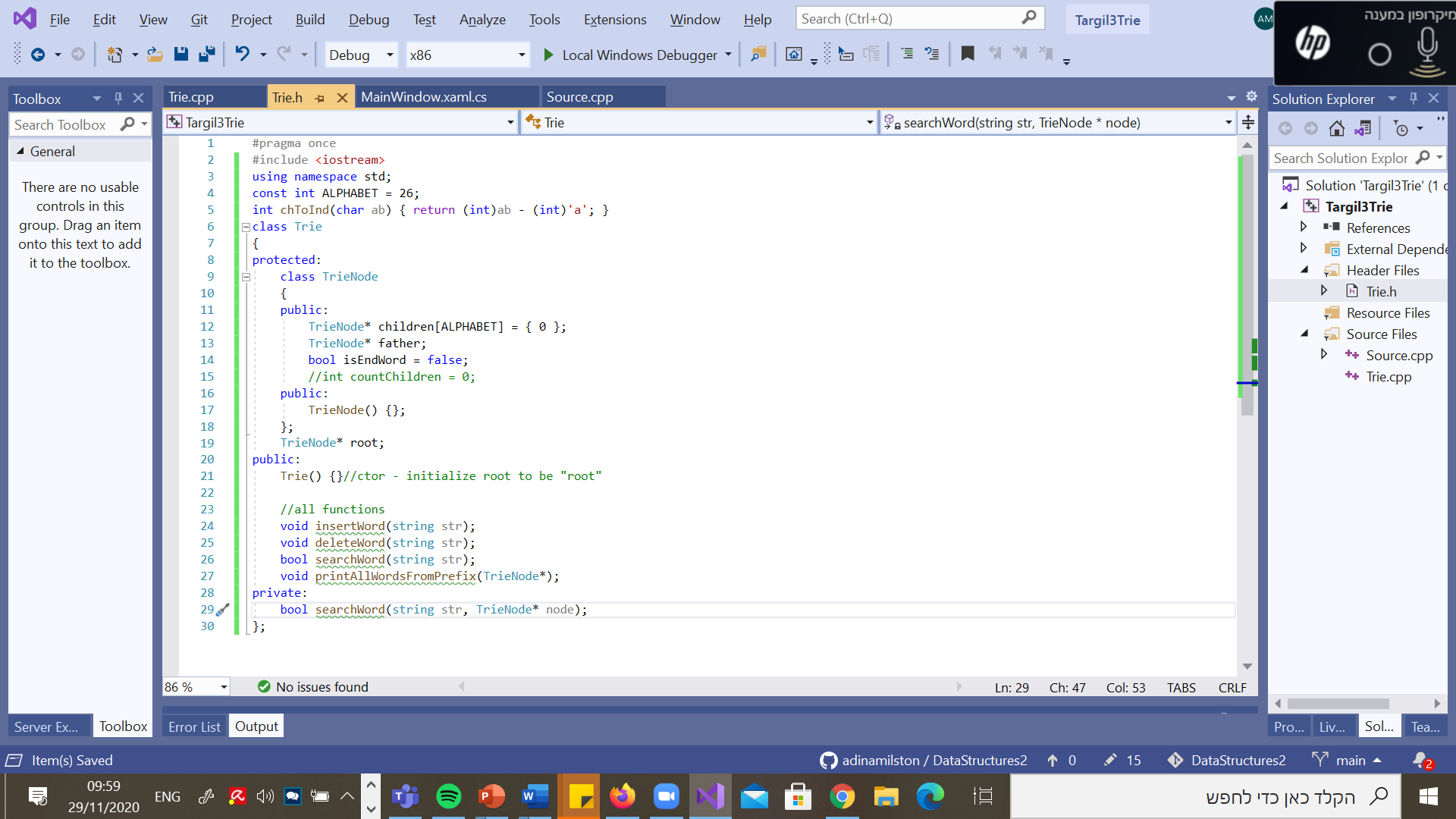
}//switch

} while (ch != '5');

return 0;

}

להלך **הצעה** למחלקות בצורה כללית:



**בהצלחה רבה!**