תיעוד חיצוני-את"מ 3

**סהר כהן: 206824088**

**יובל נהון: 206866832**

תכנית זו מקבלת תו ראשון ותו סופי אשר מגדירים (בעזרת התווים שבערכי ה ASCII שבניהם) א"ב. בנוסף, התוכנית מקבלת תו מפריד Delim, משפט קלט Sent, מילון Dict, אורך מילון DictLen ואורך מקסימאלי MaxLen. מטרת התוכנית היא השלמה אוטומטית של משפט הקלט Sent, ושמירת ההשלמה הזו במחרוזת הפלט בכתובת Output המיועדת לה. מחרוזת ההשלמה האוטומטית צריכה לענות על הדרישות הבאות:

1. Sent היא תחילית מושלמת שלה, עד כדי X טעויות (כאשר טעות יכולה להיות השמטה, הוספה או החלפה של אות).
2. היא משפט חוקי ביחס למילים המוגדרות במילון הנתון, ולמפריד Delim.

בנוסף, מחרוזת הפלט הרצויה תהיה זו בעלת האורך המקסימאלי (שאינו גדול ממש מ MaxLen), והקטנה ביותר בסדר לקסיקוגרפי מבין כל התוצאות האפשריות להשלמות.

מימוש:

התוכנית מתחילה באתחול המחסנית. לאחר מכן מאתחלים את ערכי הרגיסטרים בשימוש:

r0 – הכתובת של מחרוזת הקלט Sent.  
r1 – הכתובת של Output אשר מוקצית להכיל מחרוזת בגודל הדרוש בעת סיום ריצת התוכנית.  
r2 – הערך X אשר מייצג את מספר הטעויות המקסימאלי המותר בהרצה זו עבור תחילית.  
r3 – האורך המקסימאלי של משפט הפלט הרצוי.  
r4 – מקבלת ערך זה גם כן לצורכי השמה מחודשת ב r3 לאורך הקוד.

אם קיבלנו כקלט מילה באורך מקסימלי 0 אזי סיימנו.אחרת, ניצור מחרוזת בכתובת output אשר מורכבת אך ורק מהתו הקטן ביותר באלף בית הנתון ולבסוף @.אורך המחרוזת הוא האורך המקסימלי המבוקש.

כעת נבדוק את תקינות המשפט שהרכבנו בOutput:

תחילה נשתמש בשגרה isPrefix על מנת לבדוק האם sent היא תחילית של המשפט Output(עד כדי X טעויות-הוספה, השמטה, החלפה של אות). אם לא, נמשיך למילה הבאה(על כך בהמשך). אחרת נבדוק האם המשפט Output מהווה משפט חוקי ביחס למילון הנתון בעזרת שימוש בשגרה sentInDict. אם כן, אז סיימנו. אחרת, נעבור למילה הבאה(כמו מקודם):

מעבר למילה הבאה מתבטא בשימוש בשגרה nextString אשר תקדם את המחרוזת היושבת ב-Output למילה הבאה בא"ב הנתון.

נשתמש בשגרה isStart על מנת לבדוק אם הגענו חזרה למילה הראשונה בא"ב באורך הנתון. אם כן- אזי שעברנו על כל הקומבינציות האפשריות עבור האורך הזה ולא מצאנו התאמה, ולכן נקטין את האורך הדרוש ב-1, נאפס מחדש את המילה Output ונחזור על כל התהליך. אחרת, אם עוד לא הגענו חזרה למשפט הראשון בא"ב – נחזור על הבדיקות של isPrefix ו-sentInDict עם המילה הבאה השמורה בOutput.

בסיום תהליך זה, התוכנית תעצור בפעם הראשונה בה ניתקל במשפט חוקי (ביחס לדרישת התחיליות וביחס לשייכות למילון). וזו גם המחרוזת הקטנה ביותר בסדר לקסיקוגרפי באורך זה העונה על הדרישות – בדיוק כפי שדרוש!

**isPrefix**- שגרה שמקבלת כפרמטרים את הכתובת של מילה ושל תחילית ומספר הטעויות המותר במחסנית. היא מחזירה האם התחילית היא תחילית של המילה הנתונה עם מספר הטעויות המותר.

השגרה הינה שגרת מעטפת לשגרה רקורסיבית שנקראת auxIsPrefix. השגרה מחזירה את התוצאה באוגר r1.

**auxIsPrefix**- שגרה רקורסיבית המוכלת בתוך isPrefix. השגרה מקבלת ערכים במחסנית ובודקת התאמה עבור כל אחת מהאפשרויות לטעויות – התאמה מלאה, השמטה , הוספה.

תנאי העצירה הם:

1. המצביע לתחילית מצביע לסוף התחילית. במצב זה התחילית היא בהחלט תחילית של המילה.

2. מספר הטעויות הוא קטן מ0. במצב הזה התחילית היא לא תחילית של המילה.

בכל מקרה של זיהוי אפשרות להתאמה בין אות בתחילית לאות במילה נקרא לפונקציה הרקוריסיבית ונוריד 1 בהתאם אם היתה חוסר התאמה שהתבטאה באמצעות אחת מהאפשרויות.

**isStart**- שגרה שבודקת האם מחרוזת מכילה אך ורק את האות הראשונה בא"ב. השגרה מקבלת פרמטרים inline על ידי אוגר קישור r5. היא מקבלת את אורך המילה ואת כתובתה.

את התוצאה היא מחזירה במחסנית.

**nextString**- שגרה אשר מקדמת מחרוזת למחרוזת הבאה לפי הא"ב. אוגר הקישור שלה הוא pc. היא מקבלת את הפרמטרים שלה במחסנית. הפרמטרים שהיא מקבלת הם אורך המחרוזת וכתובתה.

השגרה מוסיפה לכתובת המחרוזת את אורכה ובכך מקבלת את האות האחרונה במחרוזת. היא רצה בסדר הפוך. אם היא נתקלת ב-Delim היא הופכת אותה לSetmin. אם היא נתקלת ב-Setmax היא הופכת אותה לDelim מתקדמת לאות הבאה ובודקת אותה. אם היא לא נתקלה באף אחת מהן היא פשוט מקדמת את האות לאות הבאה לפי סדר האותיות בטבלת ה-ascii.

**wordsEqual**- שגרה שבודקת האם שתי מחרוזות זהות זו לזו על ידי השוואה אות-אות. הפרמטרים לפונקציה הם מצביעים ל2 המחרוזות. הפונקציה מקבלת את הפרמטרים שלה במחסנית. הפונקציה מחזירה שהמחרוזות שוות אם האותיות שהושוו זהות ושוות ל’@’.

**sentInDict** – שגרה אשר מקבלת משפט, מילון, אורך מילון ותו-מפריד, ומחזירה (על המחסנית, בכתובת אשר הוקצתה מיד לפני שאר הפרמטרים לפונקציה) 1 אם המשפט הנתון חוקי ביחס למילון הנתון והמפריד, ו – 0 אחרת. משפט "חוקי" ביחס למיון הוא משפט שבו כל מילה המופרדת בעזרת התו המפריד היא מילה אשר קיימת במילון. אופן מימוש השגרה:

כאשר נגיע לסיומת המחרוזת '@' – נעצור ונחזיר 1.  
אם התווים הראשונים במשפט הם תווים מפרידים – נמשיך עד שנגיע לאות הראשונה במשפט. עבור כל אחת מהמילים במשפט, נקדם תחילה כל אחת לסופה (עד לתו המפריד או @) על מנת לחשב את אורך המילה, ואז נחזיר את המצביע חזרה לתחילת המילה הנוכחית. כעת, נשתמש בשגרה wordsEqual אשר מימשנו קודם לכן: נקצה מקום במחסנית לערך ההחזרה ונדחוף לאחר מכן את המילה הנוכחית אשר אנו בודקים. לאחר מכן, נרוץ בלולאה על **כל** מילות המילון ונפעיל את השגרה עם כל אחת מהן על מנת למצוא התאמה. כאשר מצאנו התאמה, או לאחר שסיימנו לעבור על כל מילות המילון ללא התאמה, לא נשכח גם להוציא מהמחסנית את הפרמטרים אשר דחפנו פנימה. נחזור על תהליך זה עד שנגיע למילה האחרונה במשפט הנתון (למען נוחות המימוש – נשתמש בדגל השמור ב r4 על מנת לייצג האם המילה הבאה היא המילה האחרונה במשפט), ואם אכן הגענו ל – '@' כך שלכל מילה במשפט הקלט נמצאה התאמה במילון – נחזיר 1 כרצוי. אחרת, עבור כל מילה ובה לא מצאנו התאמה – נחזיר מיד (לאחר הוצאת האיברים מהמחסנית) 0.