

## מערכות הפעלה – תרגיל 1 (תרגיל הכנה ורענון)

(לא להגשה)

התוכנית lexcmp להלן משווה לקסיקוגרפית (מילונית) בין זוג של מחרוזות. התוכנית מקבלת בשורת הפקודה שתי מחרוזות ומחזירה כערך החזרה של התוכנית ערך כדלהלן:

- 1 : אם המחרוזת הראשונה נמצאת במילון לפני המחרוזת השנייה
- 2 : אם המחרוזת השנייה נמצאת במילון לפני המחרוזת הראשונה
- 0 : אם המחרוזת הראשונה זהה למחרוזת השנייה
- -1 : במקרה של בעייה

קוד המקור לתוכנית הינו (הקובץ lexcmp.c נמצא באתר הקורס לצד התרגיל):

```
#include <string.h>

int mystrcmp(const char *str1, const char *str2)
{
    int val;

    val = strcmp(str1, str2);
    if (val < 0)
        return 1;
    if (val > 0)
        return 2;
    return 0;
}

int main(int argc, char *argv[])
{
    if (argc != 3)
    {
        return -1;
    }
    return mystrcmp(argv[1], argv[2]);
}
```

בתרגיל הינכם מתבקשים לממש, בשפת C ובסביבת אוּבונטו, תוכניות "דומות":

- את התוכנית `lencmp` אשר משווה בין זוג מחרוזות שמתקבלות בשורת הפקודה לפי אורכן ומחזירה 1 אם המחרוזת הראשונה קצרה יותר, 2 אם המחרוזות השנייה קצרה יותר, 0 אם אורך שתי המחרוזות זהה ואת הערך 1- במקרה של בעייה.
- את התוכנית `digcmp` אשר משווה בין זוג מחרוזות שמתקבלות בשורת הפקודה לפי מספר הספרות (התווים '0' עד '9') בכל מחרוזת ומחזירה 1 אם במחרוזת הראשונה יש יותר ספרות (יש להתעלם מתווים שאינם ספרות), 2 אם במחרוזות השנייה יש יותר ספרות, 0 אם בשתי המחרוזות יש מספר זהה של ספרות ואת הערך 1- במקרה של בעייה.

לשם פשטות יש ליצור קובץ `makefile` יחיד שכאשר הוא מופעל נוצרות כל שלוש התוכניות הנ"ל.

כלומר, בסופו של דבר אמורים להיות לכם 3 קבצי C כאשר בכל אחד מהם פונקציית `main` (ולכן זו תוכנית שעומדת בפני עצמה) ופונקציה/פונקציות נוספות לבחירתכם.

בנוסף יהיה לכם קובץ `makefile` יחיד אשר יגרום לקומפילציה של שלושת קבצי ה-C ובסיומו יתקבלו 3 קבצי הרצה שונים (אחד לכל תוכנית). שימו לב שקובץ ה-`makefile` ייכתב בצורה מעט שונה ממה שלמדתם מכיוון שקבצי ה-`makefile` אותם למדתם יצרו רק תוכנית אחת בסיום הרצתם ואילו כעת עליכם ליצור שלוש תוכניות בבת אחת.