

# תכנות מתקדם

## תרגיל 1

### עורך – Editor

מועד אחרון להגשה: 21.3.2021

מטרות התרגיל: vector, string

בתרגיל נממש חלקית את העורך ed (עורך ללא ממשק גרפי) המצורף ללינוקס.

#### הנחיות הגשה

- הגשת התרגיל באתר:

<http://submit.org.il/ariel>

שמות המשתמשים בקובץ login.pdf הנמצא בתיקיית תרגיל 1.

- יש להוריד מאתר מודל את הקבצים הנמצאים בתיקיית תרגיל 1 למערכת לינוקס.
- אם העורך ed לא מותקן במערכת הלינוקס, כדאי להתקינו אך לא הכרחי.

#### שיר 1

הריצו את הפקודה:

ed months.txt

הפקודה מפעילה את העורך ed ומאתחלת את מסמך העריכה על פי הקובץ months.txt.

לעורך ed אין סמן, אבל הוא ממתין לפקודות.

הקובץ input\_months.txt מכיל את כל פקודות העריכה הדרושות כדי ליצור את שיר 1.

הכניסו את הפקודות שורה אחר שורה. (אם קיבלתם ? – טעיתם בהקלדה)

העורך שמר את השיר בקובץ poem.txt, הקובץ צריך להיות זהה לקובץ month\_poem.txt.

## שיר 2

הריצו את הפקודה :

ed

ללא ארגומנט.

הקובץ input\_try.txt מכיל את כל פקודות העריכה הדרושות כדי ליצור את שיר 2. הכניסו את הפקודות שורה אחר שורה. (אם קיבלתם ? – טעיתם בהקלדה) העורך שמר את השיר בקובץ poem.txt, הקובץ צריך להיות זהה לקובץ try\_poem.txt.

## טיפ

במקום להקליד את הפקודות, אפשר לכתוב בשיר 2

ed < input\_try.txt

ובשיר 1 להוסיף שורה ראשונה

e months.txt

ולכתוב

ed < input\_months.txt

## המטלה

- יש לכתוב תכנית עורך שעבור הקלטים הקודמים תיתן את אותו פלט.
- יש להגיש חמישה קבצים : Editor.h, Editor.cpp, Document.h, Document.cpp, main.cpp
- בזמן שבוחרים את קובץ ההגשה לאתר, במקום לבחור כל קובץ בנפרד, אפשר לבחור את כל הקבצים, לחלופין אפשר להגיש קובץ זיפ של הקבצים.
- אין לשלוח makefile הוא נמצא במערכת.

## תיאור התרגיל

יש לכתוב שתי מחלקות, מחלקה Editor ומחלקה Document.

המחלקה Editor משמשת כממשק עבור המשתמש, המחלקה Document מכילה את המסמך ומממשת את פונקציות העריכה.

המחלקה Editor מכילה אובייקט של Document.

המסמך שאותו עורכים במחלקה Document ימומש כ- vector של שורות, כל שורה היא string.

למחלקה Editor יש מתודה loop() שמחכה לקלט מהמשתמש ומפעילה את המתודה המתאימה של Document.

הקובץ main.c מכיל פונקציה main שיוצרת אובייקט Editor ומפעילה את loop().

אם העורך מאותחל מקובץ אזי יופעל בנאי שיקרא את שורות הקובץ ל- vector.

להלן דוגמה:

```
#include "Editor.h"

int main(int argc, char* argv[]) {
    switch (argc) {
        case 1:
            Editor editor;
            break;
        case 2:
            . . .
    }

    editor.loop();
    return 0;
}
```

## פקודות עריכה שצריך לממש

הערות :

- מספר השורה הראשונה הוא 1
- העורך זוכר את מספר השורה האחרונה שבה התבצעה פעולת עריכה, שורה זו הופכת להיות השורה הנוכחית.
- כל פקודה מתחילה בתוו אחר ולכן נוח לממש על ידי switch.

- 3** עבור לשורה 3, השורה הנוכחית היא כעת 3.
- +2** התקדם 2 שורות קדימה, השורה הנוכחית היא כעת 5.
- 1** חזור שורה אחת אחורה, השורה הנוכחית היא כעת 4.
- \$** עבור לשורה האחרונה, השורה הנוכחית היא כעת האחרונה.

**a** הוסף שורה אחת או יותר **אחרי** השורה הנוכחית.

**שורה אחת או יותר**

. נקודה

**i** הוסף שורה אחת או יותר **לפני** השורה הנוכחית.

**שורה אחת או יותר**

.

**c** **החלף** את השורה הנוכחית בשורה אחת או יותר.

**שורה אחת או יותר**

.

**d** מחק את השורה הנוכחית.

**/text/** חפש קדימה החל מהשורה הבאה (אחרי הנוכחית) שורה שמכילה את המחרוזת text.  
אם לא מצאת, עבור לשורה 1 והמשך לחפש קדימה עד (כולל) לשורה הנוכחית.  
השורה הראשונה שנמצאה בחיפוש הופכת להיות השורה הנוכחית.

**s/old/new/** החלף בשורה הנוכחית את המופע **הראשון** של המחרוזת old במחרוזת new.

**j** חבר את השורה הנוכחית עם השורה שאחריה.

**w file** כתוב את המסמך לקובץ.

**q** צא מהעורך

**בהצלחה**