

תרגיל מסכם בלינוקס

מטרה:

- פקודות מערכת ההפעלה
- בקרת זרימה flow control
- אופרטורים
- פונקציות
- קלט/פלט לקבצים
- חשיבה אלגוריתמית
- אוטומציה (משימה 2)

משימה 1

תאור כללי:

בתרגיל זה תנהלו אוסף תקליטים.
לכל תקליט יש שם ייחודי, וייתכן יותר מעותק יחיד.

כדי לנהל את האוסף יש לממש את הפעולות הבאות

1. הוספת רשומה
2. מחיקת רשומה
3. חיפוש על פי שם או חלק ממנו
4. עדכון שם
5. עדכון כמות
6. הדפסת כמות כוללת של תקליטים (סיכום של כמות העותקים)
7. הדפסת האוסף באופן ממוין
8. יציאה מסודרת

הערות:

- יש לתכנן את התוכנה מראש באופן מודולרי, שימו לב לנקודות הבאות:
 - שימוש חוזר בקוד (בעזרת פונקציות)
 - תשומת לב לסקופינג - מרחבי ההיכרות של משתנים (לוקליות וגלובליות)
 - תכנון ממשקים - איך פונקציות "מדברות" אחת עם השנייה.
 - נוחות תחזוקה ושינוי של הקוד
 - תיעוד סביר
 - יש להקפיד על שמות פונקציות ומשתנים ברורים ואינפורמטיביים
- כל פעולה תמומש כפונקציה נפרדת ויש לתכנן את התוכנה כולה בחלוקה לפונקציות ופונקציות עזר.
 - יש לממש בעזרת כמה שיותר פונקציות עזר.
 - האוסף יירשם לקובץ טקסטואלי בעל מבנה טבלאי של שני שדות המופרדים ביניהם בפסיק:
שימו לב שימו לב, קובץ הרשומות ייבדק באופן אוטומטי יש לוודא שהמבנה שלו זהה בדיוק למבנה שצוין פה!

`RECORD_NAME, AMOUNT`

- שם הקובץ ששומר על הנתונים יינתן כארגומנט חיצוני בשורת הפקודה
`scriptName recordFileName`
- כל פעולה שתבוצע על האוסף תלווה ברישום לקובץ לוג. שימו לב שימו לב, הלוג ייבדק באופן אוטומטי יש לוודא שהמבנה שלו זהה בדיוק למבנה שצוין פה!

- יש לבדוק קלט ולהציג הודעות שגיאה על קלט לא תקין (לדוג'):
 - בדיקות קלט של הארגומנטים החיצוניים (יציאה עם הודעה המסבירה על הפעלת התוכנית)
 - אופציות תפריטים, יש להציג שגיאה ובקשה לקלט חוזר
 - השדה שם תקליט יכול להכיל **אותיות מספרים ורווחים** בלבד
 - השדה כמות תקליטים יכול להכיל מספרים בלבד, שלמים וחיוביים.
- יש לשים לב לממשק משתמש
 - שפה ברורה וידידותית למשתמש
 - מענה לכל הפונקציונליות הנדרשת (כולל יציאה מסודרת)
 - פלטים על הצלחת פעולה ושגיאה באופן ובשפה שמובנים לבני אנוש.

תכנון:

מומלץ לתכנן פונקציות ראשיות מסוימות לפני פונקציות ראשיות אחרות. מומלץ לחלק פונקציות ראשיות לתתי פונקציות (פונקציות עזר) עפי' זיהוי התנהגות מסוימת משותפת. כאמור יש לממש כמה שיותר פונקציות עזר. יש לתכנן את פונקציות העזר כפונקציות בעלות תפקיד אחד ויחיד ולהמנע מפונקציות כלבו שיודעות לעשות הכל.

בעת תכנון פונקציות יש לחשוב על הנקודות הבאות :

- מהו תפקיד הפונקציה.
- מה הקלט שלה – האם הוא מועבר באמצעות ארגומנטים או קלט משתמש.
- מה ה"אפקט"ים של הפונקציה (שמירה לקובץ, הצגת פלט למשתמש וכו')
- מהם הערכים המוחזרים של הפונקציה.

יש לתכנן כמה שיותר על דף נייר בעזרת תרשימי זרימה ופסאודו קוד (מצופה מכם להציג את סקיצות התכנון שלכם). מומלץ דף ועט ואין צורך להקפיד על אסתטיקה ונראות של השרטוטים. התכנון שלכם צריך להציג

- את התנהלות התוכנית ברמה הגבוהה יותר (אלגוריתם התפריט)
- התנהלות עבור כל אופציה בתפריט (אלגוריתם עבור כל פונקציה ראשית).
- התנהלות כל תת פונקציה. (אלגוריתם עבור כל פונקציית עזר)
- יש להציג את העברת המידע אל ומתוך כל פונקציה (קלט פלט של פונקציה יכול להיות פרמטרים וערך מוחזר וגם קלט מהמשתמש)
- בעת תכנון האלגוריתם עבור כל פונקציה יש לחשוב על הנקודות הבאות
 - האם סדר ההוראות לפתרון הבעיה נכון.
 - האם ההוראות הן חד משמעיות.
 - האם הוגדר קלט/פלט באופן נכון.
 - האם הוגדרה נקודת סיום ברורה לאלגוריתם.

פונקציות ראשיות:

שימו לב - מומלץ לממש פונקציות ראשיות מסוימות לפני פונקציות ראשיות אחרות.

1. הוספת רשומה. פרמטרים: שם התקליט, כמות

פונקציה זו תוסיף רשומה או תעדכן רשומה קיימת

- אם יש יותר מתקליט אחד העונה על תוצאת החיפוש, יוצג תפריט ממוספר עם שמות תוצאות המועמדים, ובהקלדת מספר מתאים ייבחר התקליט התואם להוספה. (שימו לב לתפריט המוצע תהיה גם אופציה להוסיף את ערך החיפוש עצמו כתקליט חדש)
- במקרה ורשומה אינה קיימת כלל, תתווסף שורה לסוף הקובץ במידה ותקליט אינו קיים (רמז: חיפוש לא החזיר תוצאה)
- במידה וקיים יתבצע עדכון כמות התקליטים ברשומה הרלוונטית (ללא כתיבה מחדש של הקובץ)
- בסיום תוצג הודעה על הצלחת המשימה, רישום בקובץ לוג וחזרה לתפריט ראשי.

2. מחיקת רשומה. פרמטרים: שם התקליט, כמות

פונקציה זו תסיר רשומה או תעדכן רשומה קיימת

- אם יש יותר מתקליט אחד העונה על תוצאת החיפוש יוצג תפריט ממוספר עם שמות תוצאות המועמדים ובהקלדת מספר מתאים, ייבחר התקליט התואם למחיקה.
- במידה ונשארו עותקים יתבצע עדכון כמות התקליטים ברשומה הרלוונטית (ללא כתיבה מחדש של הקובץ)
- במידה ולא נשארו עותקים תתבצע מחיקה של הרשומה ללא הותרת שורה רווח.
- תינתן הודעת שגיאה במידה ותקליט אינו קיים (רמז: חיפוש לא החזיר תוצאה)
- תינתן הודעת שגיאה במידה והכמות חורגת מהקיים.
- בסיום תוצג הודעה על הצלחת או כישלון המשימה, רישום בקובץ לוג וחזרה לתפריט ראשי.

3. חיפוש על פי שם או חלק ממנו. פרמטרים: מחרוזת חיפוש.

- פונקציה זו תבצע חיפוש בקובץ ותציג למשתמש רשימה של תקליטים המכילים את מחרוזת החיפוש.
- במידה ויש שם תקליט אחד או יותר, תוצג רשימה ממוינת לקסיקוגרפית (סדר מילוני) של כל התוצאות האפשריות כולל כמויות.
- במידה ואין תוצאת חיפוש יש להציג הודעה על כישלון החיפוש.
- בסיום יתבצע רישום בלוג וחזרה לתפריט הראשי.

4. עדכון שם. פרמטרים: שם ישן, שם חדש.

פונקציה זו תעדכן שם של תקליט ברשומה קיימת

- אם יש יותר מתקליט אחד העונה על תוצאת החיפוש, יוצג תפריט ממוספר עם שמות תוצאות המועמדים ובהקלדת מספר מתאים ייבחר התקליט התואם לעידכון
- תינתן הודעת שגיאה במידה ותקליט אינו קיים (רמז: חיפוש לא החזיר תוצאה)
- בסיום תוצג הודעה על הצלחת או כישלון המשימה, רישום בלוג וחזרה לתפריט ראשי.

5. עדכון כמות. פרמטרים: שם, כמות

פונקציה זו תעדכן כמות של תקליטים ברשומה קיימת

- אם יש יותר מתקליט אחד העונה על תוצאת החיפוש, יוצג תפריט ממוספר עם שמות תוצאות המועמדים ובהקלדת מספר מתאים ייבחר התקליט התואם לעידכון.
- תינתן הודעת שגיאה במידה ותקליט אינו קיים (רמז: חיפוש לא החזיר תוצאה)
- תינתן הודעת שגיאה על כמות שקטנה מ 1
- בסיום תוצג הודעה על הצלחת או כישלון המשימה, רישום בלוג וחזרה לתפריט ראשי.

6. **הדפסת כמות כוללת של תקליטים.** פרמטרים: אין
- פונקציה זו תסרוק את הקובץ ותסכם את כמות כלל התקליטים במאגר (כמות עותקים כוללת)
- במידה ופעולת הסיכום עברה בהצלחה וגדולה מאפס, יש להציג הודעה למשתמש על כמות התקליטים.
 - אם אין תקליטים במאגר תוצג הודעה מותאמת ושונה
 - בסיום יתבצע רישום בלוג הכולל את התוצאה וחזרה לתפריט ראשי.

7. **הדפסת האוסף באופן ממוין.** פרמטרים: אין
- פונקציה זו תדפיס את הקובץ במיון לקסיקוגרפי (סדר מילוני) על פי שמות התקליטים
- במידה ויש תקליטים במאגר יודפסו שמות וכמויות של כל המאגר באופן ממוין
 - אם אין תקליטים במאגר תוצג הודעה מותאמת ושונה
 - בסיום יתבצע רישום בלוג הכולל את התוצאה וחזרה לתפריט ראשי.

8. **כתיבה ללוג.** פרמטרים : האירוע שקרה והאם הסתיים בהצלחה או כישלון וערכים רלוונטיים להדפסה
- פונקציה זו תדאג לכתוב לקובץ לוג את האירוע הנוכחי.
- שימו לב שימו לב, הלוג ייבדק באופן אוטומטי יש לוודא שהמבנה שלו זהה בדיוק למבנה שצוין פה!
 - שם הקובץ יהיה מורכב משם קובץ הרשומות קו תחתון והמילה לוג
recordFileName_log
 - כל שורה בלוג מציינת אירוע אחד, והיא כוללת תאריך, שעה, שם אירוע והאם הסתיים בהצלחה (למעט אירועי הדפסה)
דוגמה:

- `28/2/2022 16:30:50 Insert Success`
 - הצלחה/כישלון
 - i. Failure
 - ii. Success
- שם האירוע:
 - i. Insert
 - ii. Delete
 - iii. Search
 - iv. UpdateName
 - v. UpdateAmount
 - vi. PrintAmount
 - vii. PrintAll

- אירועי הדפסה:
 - i. עבור הדפסת כמות יודפס שם האירוע וכמות התקליטים (כולל 0)
- `28/2/2022 16:50:47 PrintAmount AMOUNT`
 - i. עבור הדפסת האוסף יודפסו בלוג שורות כמספר התקליטים:
- `28/2/2022 16:47:52 PrintAll RECORD_NAME1 AMOUNT`
- `28/2/2022 16:47:52 PrintAll RECORD_NAME2 AMOUNT`
- `28/2/2022 16:47:52 PrintAll RECORD_NAME3 AMOUNT`
- `28/2/2022 16:47:52 PrintAll RECORD_NAME4 AMOUNT`

משימה 2

יש לכתוב תהליך אוטומטי הבוחן את הלוג מדי יום ושולח מייל עם הלוג, במידה והיו שגיאות באחד מהאירועים.

הסקריפט יקבל כארגומנט חיצוני את שם הלוג, ואחת ל-24 שעות יבדוק האם היו שגיאות בלוג, במידה וכן יישלח מייל הכולל את הלוג לכתובת מייל המוגדרת במשתנה סביבה.

פרנסים:

איך לוודא שאנחנו ב character class

```
#!/usr/local/bin/bash
if [[ $1 =~ [:digit:] ]]; then
    echo "true"
else
    echo "False"
fi
```

• מהם ה classes:

ASCII letters and digits — [:alnum:]

ASCII letters — [:alpha:]

ASCII characters — [:ascii:]

space and tab :ASCII widthful whitespace — [:blank:]

ASCII 0 to 31 :control" characters" — [:cntrl:]

d\ same as ,ASCII digits — [:digit:]

ASCII characters that use ink — [:graph:]

ASCII lower-case letters — [:lower:]

ASCII ink-users plus widthful whitespace — [:print:]

s\ same as ,ASCII whitespace — [:space:]

ASCII upper-case letters — [:upper:]

w\ same as ,_ ASCII letters and — [:word:]

ASCII hex digits — [:xdigit:]

החלפות תוכן בתוך קובץ בעזרת : sed

<https://www.cyberciti.biz/faq/how-to-use-sed-to-find-and-replace-text-in-files-in-linux-unix-shell/>

<https://www.folkstalk.com/2013/03/sed-remove-lines-file-unix-examples.html>

פירמוט תאריכים:

<https://www.geeksforgeeks.org/date-command-linux-examples/>