

יטודות מערכות פתוחות
תרגיל מס' 11

מועד ההגשה האחרון להגשת התרגיל מופיע באתר הקורס

שימו לב: כל ההערות שבתחילת תרגילים 10-1 תקפות גם לתרגיל זה.

1. שאלה זו הופיעה במבחן מועד ב 2017

בחברה להשכרת רכבים שומרים נתונים על השכרות בקובץ ששמו `rents`
שכל שורה שלו היא במבנה הבא:

תאריך החזרה : תאריך השכרה : דגם הרכב : שם השוכר : מחיר ליום
עבור רכב שהושכר ולא הוחזר מופיע - (מינוס) בשדה של תאריך
ההחזרה.

לדוגמה השורה הבאה:

120:Yossi Levi:Toyota 2008:1/1/2016:22/1/2016

מצינת שרכב מדגם טויוטה 2008 הושכר על ידי יוסי לוי החל
מתאריך 1/1/2016 ועד תאריך 22/1/2016 (כולל), במחיר של 120
ש"ח ליום.

כתוב/כתבי תכנית ב- `awk` בשם `P11.1` שמקבלת כפרמטרים שם שוכר
ושני תאריכים, ומדפיסה לפלט בשורה הראשונה את שם השוכר
והתאריכים המבוקשים, ובשורות שלאחר מכן את כל השורות שמתארות
השכרות של השוכר שנמצאות בתחום שבין שני התאריכים שהועברו
כפרמטרים לתכנית. (תיתכן חפיפה חלקית לתחום כפי שמתואר בדוגמה
שבהמשך). בסוף כל שורה יופיע תו רווח ולאחריו מספר שמציין
את המחיר הכולל של ההשכרה בתחום שמתקבל מהחפיפה בין התחום
שמתואר על ידי השורה לבין התחום שמתואר על ידי התאריכים
המבוקשים (כפי שמתואר בדוגמה שבהמשך).

יש להניח שהתאריך השני שמועבר לתכנית קטן או שווה לתאריך שבו
מופעלת התכנית.

אין להשתמש בפקודות `system` בתכנית `P11.1`.

על סדר שורות הפלט להיות לפי סדר הופעתן בקובץ `rents`.

לדוגמה אם השוכר Yossi Levi שכר את הרכב Toyota 2008 החל מתאריך 1/1/2016 ועד תאריך 20/1/2016 (כולל) במחיר של 120 ש"ח ליום, ואם התאריכים המבוקשים הם מתאריך 11/1/2016 ועד תאריך 22/1/2016, אז בסוף השורה שמתארת את ההשכרה הזו יופיע המספר 1200 כי החפיפה בין התחומים מכילה בדיוק 10 ימים, דהינו החל מ- 11/1/2016 ועד 20/1/2016.

לצורך הפתרון של השאלה מומלץ להשתמש ב- 2 הפונקציות הבאות:

הפונקציה: `is_date_less_than(date1,date2)` שנודרשתם לעשות בתרגיל בית 10.2.

הפונקציה: `number_of_days_in(date1,date2)` שמחזירה מספר שמתאר את מספר הימים שנמצאים בין התאריכים `date1` ו- `date2` כולל `date1` ו- `date2`

לדוגמה הקריאה:

```
number_of_days_in(01/01/2016,20/01/2016)
```

מחזירה 20.

הקריאה:

```
number_of_days_in(01/01/2016,01/01/2016)
```

מחזירה 1.

בהמשך מתואר מימוש של הפונקציה הזו (ניתן להעתיק את המימוש הזה ולהשתמש בו בתכנית שלכם).

לדוגמה, נניח שתוכן הקובץ `rents` הוא:

```
120:Yossi Levi:Toyota 2008:1/1/2016:20/1/2016
100:Dan Cohen:Nissan 1998:2/01/2015:03/01/2015
400:Yossi Levi:Toyota 2015:11/1/2015:12/1/2016
200:Yossi Levi:Toyota 2010:1/1/2015:20/1/2015
300:Yossi Levi:Toyota 2016:18/1/2016:-
```

לאחר הפעלת התכנית על ידי הפקודה:

```
P11.1 "Yossi Levi" 11/1/2016 22/1/2016
```

מתקבל הפלט:

```
Yossi Levi 11/1/2016 22/1/2016
120:Yossi Levi:Toyota 2008:1/1/2016:20/1/2016 1200
400:Yossi Levi:Toyota 2015:11/1/2015:12/1/2016 800
300:Yossi Levi:Toyota 2016:18/1/2016:- 1500
```

שימו לב שהשורה:

200:Yossi Levi:Toyota 2010:1/1/2015:20/1/2015

אינה מופיעה בפלט כי אין חפיפה בין התחום המבוקש של התאריכים לתחום התאריכים של שורה זו.

שימו לב שבגלל שבשורה:

300:Yossi Levi:Toyota 2016:18/1/2016:- 1500

יש מינוס "-" בתאריך ההחזרה, ההנחה היא שעד התאריך 22/1/2016 לא הסתיימה תקופת ההשכרה. ולכן החישוב של המחיר הכולל (דהינו 1500) עבור השורה הנ"ל התבצע עבור 5 ימים החל מ- 18/1/2016 ועד 22/1/2016.

להלן תכנית בשם P1 שמכילה מימוש של הפונקציה number_of_days_in

(ניתן להעתיק את המימוש הזה של הפונקציה לתכנית שלכם).

לדוגמה לאחר הקריאה לתכנית:

P1 1/1/2018 3/1/2018

מתקבל הפלט:

3

לדוגמה לאחר הקריאה לתכנית:

P1 1/1/2018 1/1/2019

מתקבל הפלט:

366

להלן התכנית P1:

```
#!/bin/awk -f
function number_of_days_in(date1,date2,A,B,d1,d2){
    gsub("/"," ",date1)
    gsub("/"," ",date2)
    split(date1,A," ")
    split(date2,B," ")
    d1=A[3]" "A[2]" "A[1]" "00" "00" "00
    d2=B[3]" "B[2]" "B[1]" "00" "00" "00
    return ((mktime(d2)-mktime(d1))/86400+1)
}
BEGIN {print number_of_days_in(ARGV[1],ARGV[2])}
```

ובפורמט טקסט:

```
#!/bin/awk -f
function number_of_days_in(date1,date2,A,B,d1,d2) {
  gsub("/"," ",date1)
  gsub("/"," ",date2)
  split(date1,A," ")
  split(date2,B," ")
  d1=A[3]" "A[2]" "A[1]" "00" "00" "00
  d2=B[3]" "B[2]" "B[1]" "00" "00" "00
  return ((mktime(d2)-mktime(d1))/86400+1)
}
BEGIN {print number_of_days_in(ARGV[1],ARGV[2])}
```

2. כתוב תכנית בשם P11.2 שמבצעת את אותה המשימה של שאלה 1 אבל ב- `bash`. אין להשתמש ב- `awk` ו- `sed` בתכנית P11.2. התכנית P11.2 יכולה לקרוא לשתי תכניות עזר בשם P11.2.1 ו- P11.2.2 שמחשבות את הפונקציות `number_of_days_in` ו- `is_date_less_than`.

במילים אחרות, התכנית P11.2.1 תהיה תכנית ב- `awk` שמקבלת שני פרמטרים של תאריכים ומדפיסה YES אם הראשון קטן מהשני או NO אחרת. התכנית P11.2.2 תהיה תכנית ב- `awk` בדיוק כפי שתואר בשאלה 1 שמקבלת שני תאריכים ומדפיסה לפלט את ההפרש ביניהם).

3. שאלה זו הופיעה במבחן מועד א 2017

נגדיר שמילה היא רצף של תווים ללא תווי רווח וסוף שורה.

כתוב תכנית Script ב- `sed` (במבנה כפי שנדרש בכל שאלות ה- `sed` ותואר בפרוט בתרגיל בית 7.2) בשם P11.3 שמקבלת כפרמטר שם קובץ ומדפיסה את הפלט שמתקבל באופן הבא:
אם בשורה בקובץ יש פחות משתי מילים אז השורה לא מופיעה בפלט.

אם בשורה בקובץ במילה השניה בשורה יש פחות מ- 3 תווים אז השורה לא מופיעה בפלט.

אם בשורה בקובץ במילה השניה יש לפחות 3 תווים אז מופיעה בפלט השורה שמתקבלת לאחר שכפול 3 התווים הראשונים במילה השניה בשורה, כפי שמתואר בדוגמה שבהמשך.

בסוף הפלט מופיע מספר שמציין את מספר השורות בקובץ שלא הופיעו בפלט.

לדוגמה, נניח שתוכן הקובץ F2 הוא:

```
abc2 1 def2 3 abc2
dea 123 123 zy45
12 12
abc cd abc cd
xyz%$@! x@yz%$@! kk
xyzxyz
abc htzw
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P11.3 F2
```

יתקבל הפלט:

```
dea 112233 123 zy45
xyz%$@! xx@yyz%$@! kk
abc hhttzzw
4
```

עליכם להגיש את כל קבצי ה-script דהינו הקבצים: P11.1 – P11.3
והקבצים P11.2.1 ו-P11.2.2 לבדיקה אוטומטית ע"י הפקודה:

```
~basicsys/submit
```

מותר להגיש בזוגות אך עדיף להגיש לבד.

שימו לב שגם אם בחרתם לא להשתמש בקבצי העזר P11.2.1 ו-P11.2.2 עליכם להגיש את הקבצים האלה
(עם תוכן כלשהו) כי אחרת ההגשה של התכניות שלכם לא תתקבל.

בהצלחה !