

יטודות מערכות פתוחות  
פתרון תרגיל מס' 8

שימו לב: כל ההערות שבתחילת תרגילים 1-7 תקפות גם לתרגיל זה.

**1.** כתוב תוכנית Script ב-Bash בשם P8.1 שמקבלת כפרמטרים שם קובץ ורשימה של זוגות של מספרים מהצורה x:y ומדפיסה פלט כפי שמתואר בדוגמה שלהלן.

לדוגמה, נניח שתוכן הקובץ F1 הוא:

```
ab 865a 1122 ers
ab cd ef gh
14 25ab 33cd 778abcdx
5555 abcd
```

לאחר הפעלת התכנית על ידי הפקודה:

```
P8.1 F1 3:6 1:5 4:-8 2:4 3:7
```

מתקבל הפלט הבא:

```
1122 abers 865a 1122
ef abgh cd ef
33cd 14778abcdx25ab 33cd
5555 abcd
```

הסבר לפלט

כל שורה בפלט מכילה את המילים מהקובץ בפורמט כפי שמתואר על ידי הזוגות של הפרמטרים לתכנית. בתחילת השורה הראשונה של הפלט מופיעה המילה 1122 עם תוספת שני רווחים בצד שמאל. מילה זו מתאימה לזוג 3:6 שאומר הצג את המילה השלישית בשורה (שהיא המילה 1122) בפורמט של 6 תווים ולכן יש להוסיף 2 תווי רווח מצד שמאל. בהמשך השורה הראשונה של הפלט מופיע המילה ab אם תוספת של 3 רווחים מצד שמאל. מילה זו מתאימה לזוג 1:5 כי היא המילה הראשונה בשורה ומודפסת בפורמט של 5 תווים ולכן יש להוסיף 3 תווי רווח בצד שמאל. בהמשך השורה הראשונה מופיעה המילה ers בתוספת 5 רווחים מצד ימין. מילה זו מתאימה לזוג 4:-8 כי היא המילה הרביעית בשורה ומודפסת בפורמט של 8 תווים עם השלמת התווים החסרים מצד ימין. שימו לב שבשורה האחרונה יש 7 רווחים בתחילת השורה ו- 7 רווחים בסוף השורה.

## הנחות:

ניתן להניח שאם ברשימת הזוגות הועבר זוג מהצורה  $i:j$  אז המילה  $i$  - בכל שורה בקובץ מכילה לכל היותר  $j$  תווים (או שאינה קיימת כלל). לדוגמה בגלל שיש זוג מהצורה 3:6 ברשימת הפרמטרים לתכנית, אז ניתן להניח שבכל שורה בקובץ המילה השלישית מכילה לכל היותר 6 תווים (או שאינה קיימת כלל).

## פתרון שאלה 8.1

```
file_name=$1
shift
while read line; do
    for param in "$@"; do
        i=$(echo $param | cut -d":" -f1)
        j=$(echo $param | cut -d":" -f2)
        echo $line | awk '{printf "%'$j's",'$i'}'
    done
    echo
done <$file_name
```

להלן אותו פתרון בפורמט טקסט.

```
file_name=$1
shift
while read line; do
    for param in "$@"; do
        i=$(echo $param | cut -d":" -f1)
        j=$(echo $param | cut -d":" -f2)
        echo $line | awk '{printf "%'$j's",'$i'}'
    done
    echo
done <$file_name
```

2. כתוב תוכנית Script ב-awk בשם P8.2 שמקבלת כפרמטרים רשימת שמות קבצים ומדפיסה שורה אחת עבור כל קובץ שמכילה את שם הקובץ ולאחר מכן נקודותיים ולאחר מכן מספר שמציין את אורך המילה הארוכה ביותר בקובץ. על סדר השורות בקובץ להיות להיות לפי סדר הפרמטרים לתכנית.

לדוגמה, נניח שתוכן הקובץ F2 הוא:

```
12345678 abcdefghi aa
abc cdefg 112233 55555555
```

ונניח שתוכן הקובץ F3 הוא:

```
1234 123456
12345
234567
```

ונניח שתוכן הקובץ F1 הוא כפי שמתואר בשאלה 1. לאחר הפעלת התכנית על ידי הפקודה: P8.2 F1 F2 F3 יתקבל הפלט:

```
F1:8
F2:9
F3:6
```

שימו לב שהשורה הראשונה בתכנית P8.2 צריכה להיות:

```
#!/bin/awk -f
```

## פתרון שאלה 8.2

```
#!/bin/awk -f
BEGIN {fname=ARGV[1]}
FILENAME == fname {
    for (i=1; i <= NF; i++) {
        if (length($i) > max)
            max=length($i)
    }
}
FILENAME != fname {
    print fname":"max
    fname=FILENAME
    max=length($1)
}
END {
    print fname":"max
}
```

להלן אותו פתרון בפורמט טקסט.

```
#!/bin/awk -f
BEGIN {fname=ARGV[1]}
FILENAME == fname {
    for (i=1; i <= NF; i++) {
        if (length($i) > max)
            max=length($i)
    }
}
FILENAME != fname {
    print fname":"max
    fname=FILENAME
    max=length($1)
}
END {
    print fname":"max
}
```

3. שאלה זו הופיע במבחן מועד א 2016 (החומר לפתרון שאלה זו נמצא בחומר ללימוד עצמי כפי שמופיע באתר הקורס).

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

כתוב תכנית Script ב- sed בשם P8.3 שמקבלת כפרמטר שם קובץ ומדפיסה את הפלט שמתקבל באופן הבא:  
אם בשורה בקובץ אין שתי מילים רצופות זהות, אז השורה לא מופיעה בפלט.  
אם בשורה בקובץ יש שתי מילים רצופות זהות, אזי מודפסת לפלט השורה שמתקבלת לאחר החלפת שתי המילים הרצופות הזוהות הראשונות בשורה וכל הרווחים בינהן, במילה אחת, כפי שמתואר בדוגמה שבהמשך.

בסוף הפלט מופיע מספר שמציין את מספר השורות בקובץ שאין בהן שתי מילים רצופות זהות.

על המבנה של התכנית P8.3 להיות כדלהלן:

התכנית מכילה שורה אחת או יותר של פקודות sed לדוגמה:

```
sed s/dog/cat/ $1
```

אין להשתמש בפקודות שאינן של sed ושאינן מתחילות ב- sed.  
מותר לכתוב לקובץ ביניים כמו למשל:

```
sed s/dog/cat/ $1 >| tmp
```

מותר גם לקרוא מקובץ ביניים כמו למשל:

```
sed s/dog/cat/ < tmp
```

אסור להשתמש ב- pipeline זאת אומרת המבנה הבא אסור:

```
sed s/dog/cat $1 | sed s/abc/def
```

לדוגמה, נניח שתוכן הקובץ F4 הוא:

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

לאחר הפעלת התכנית ע"י הפקודה:

P8.3 F4

יתקבל הפלט:

```
dea 123 zy45
12
xyz%$@! kk
abc h h h
3
```

### פתרון שאלה 8.3

```
sed -r '/(^|[ ]+)(^[ ]+)[ ]+\2([ ]+|$)/!d' $1 >| tmp1
sed -r '/(^|[ ]+)(^[ ]+)[ ]+\2([ ]+|$)/d' $1 >| tmp2
sed -r 's/(^|[ ]+)(^[ ]+)[ ]+\2([ ]+|$)/\1\2\3/' tmp1
sed -n '$=' tmp2
```

להלן אותו פתרון בפורמט טקסט.

```
sed -r '/(^|[ ]+)(^[ ]+)[ ]+\2([ ]+|$)/!d' $1 >| tmp1
sed -r '/(^|[ ]+)(^[ ]+)[ ]+\2([ ]+|$)/d' $1 >| tmp2
sed -r 's/(^|[ ]+)(^[ ]+)[ ]+\2([ ]+|$)/\1\2\3/' tmp1
sed -n '$=' tmp2
```