

## תוספות להרצאה מספר 9

להלן תוספות להרצאה 9 שנדרשות לפתרון תרגיל בית מספר 9.

### דוגמה 1

התכנית P1 מקבלת 3 פרמטרים: שני מספרים j ו i ושם קובץ ומדפיסה לפלט את העמודה ה- j בקובץ מיושרת לימין כך שהאורך של כל מילה (כולל הרווחים) יהיה i.

ניתן להניח שבכל שורה בקובץ יש לפחות j מילים.

לדוגמה, נניח שתוכן הקובץ F1 הוא:

```
123  abcd  a  uu
ab  yztu  1234567
g  abcdef  abcd
abcde ghej 12
```

לאחר הפעלת התוכנית על ידי הפקודה: P1 3 7 F1

מתקבל הפלט:

```
      a
1234567
      abcd
      12
```

תוכן התכנית הוא:

```
j=$1
i=$2
file=$3
rm -f tmp1
while read line
do
    echo $line | cut -d" " -f$j >> tmp1
done<$file
cat tmp1 | awk '{printf "%'i's\n",$1}'
```

פתרון יותר קצר הוא התכנית הבאה:

```
j=$1
i=$2
file=$3
cat $file | awk '{printf "%'i's\n",$'j}'
```

## דוגמה 2

התכנית P2 מקבלת כפרמטר שם קובץ שמכיל מטריצה ריבועית ומדפיסה את סכום המספרים במסגרת החיצונית שלו.

לדוגמה, נניח שתוכן הקובץ F2 הוא:

```
1 1 1 1
1 2 2 1
1 3 2 1
1 1 1 1
```

לאחר הפעלת התוכנית על ידי הפקודה: P2 F2

מתקבל הפלט:

```
12
```

תוכן התכנית הוא:

```
#!/bin/awk -f

{ for (j=1;j <=NF; j++) {
  A[FNR,j]=$j
}
}
END { for (i=1; i<=FNR; i++) {
  for (j=1; j<= FNR; j++) {
    if ( i==1 || i==FNR || j==1 || j==FNR)
      s+=A[i,j]
  }
}
print s
}
```

### דוגמה 3

התכנית P3 ב-sed מקבלת כפרמטר שם קובץ ומדפיסה לפלט את השורות שמכילות לפחות שתי ספרות, כאשר בשורות האלה מתבצעת החלפה בין המילה הראשונה למילה האחרונה.

לדוגמה, נניח שתוכן הקובץ F3 הוא:

```
one two three
one1 two three
one2 two3 three
12345
11 22 33 444
1a 2b
```

לאחר הפעלת התוכנית על ידי הפקודה: P3 F3

מתקבל הפלט:

```
three two3 one2
12345
444 22 33 11
2b 1a
```

פתרון:

```
sed 's/[0-9].*[0-9]/!d' $1 >|tmp
sed 's/^\([ ]*\)\(\^[ ]+\)\(.*\)\(\[ ]+\)\(\^[ ]+\)\(\[ ]*\)
*\)$/\1\5\3\4\2\6/' < tmp
```

לצורך הבנת הביטוי הרגולארי נרשום אותו ללא הסימן \ גם במבחן אפשר לרשום ללא הסימן \, אך יש לציין זאת במפורש).

```
sed 's/^\([ ]*\)\(\^[ ]+\)\(.*\)\(\[ ]+\)\(\^[ ]+\)\(\[ ]*\)
$/\1\5\3\4\2\6/' < tmp
```

## פתרון נוסף:

הפתרון הבא מחלק את הבעיה ל- 2 המקרים הבאים:

מקרה 1: יש בדיוק 2 מילים בשורה עם רווח אחד בדיוק בין המילים.

מקרה 2: יש לפחות 2 מילים בשורה ומקרה 1 לא מתקיים. במקרה זה ישנו לפחות רווח אחד אחרי המילה הראשונה וישנו לפחות רווח אחד לפני המילה האחרונה.

שימו לב שבשורה שלא מתאימה למקרה 1 או למקרה 2 אין צורך לבצע החלפה.

לצורך נוחות הקריאה הפתרון מוצג ללא התו \ בביטויים הרגולאריים.

```
sed '/[0-9].*[0-9]/!d' $1 >|tmp
sed 's/^( [ ]*)( [^ ]+)( [ ])( [^ ]+)( [ ]*)$/
\1\4\3\2\5/' tmp >| tmp1
sed 's/^( [ ]*)( [^ ]+)( [ ]+.*[ ]+)( [^ ]+)( [ ]*)$/
\1\4\3\2\5/' tmp1
```

הפתרון הנ"ל בתוספת התו \ הוא:

```
sed '/[0-9].*[0-9]/!d' $1 >|tmp
sed 's/^\([ ]*\)\([^\ ]+\)\([ ]\)\([^\ ]+\)\([ ]*\)$/
\1\4\3\2\5/' tmp >| tmp1
sed 's/^\([ ]*\)\([^\ ]+\)\([ ]+.*[ ]+\)\([^\ ]+\)\([ ]*\)$/
\1\4\3\2\5/' tmp1
```