

יסודות מערכות פתוחות
פתרון תרגיל מס' 8

1. שאלה זו הופיע במבחן מועד א 2015

נגדיר שקובץ הוא טוב אם מספר המילים הגדול ביותר בשורה בקובץ שווה למספר השורות בקובץ.

לדוגמה הקובץ הבא הוא טוב:

```
ab  
ab cd ef  
123 ab
```

כתוב תוכנית Script ב-Bash בשם P8.1 שמקבלת כפרמטרים רשימת תיקיות ומדפיסה לפלט שורה אחת עבור כל תיקיה שמכילה את שם התיקיה לאחריה תו רווח אחד, לאחרי מספר שמציין את מספר הקבצים הטובים בתיקיה (בעומק כלשהו). על סדר התיקיות בפלט להיות לפי סדר הופעתן ברשימת התיקיות.

ראה/י דוגמה בעמוד הבא

לדוגמה, להלן מבנה תיקיות d1 d2 d3 כפי שמתקבל על ידי הפעלת
 הפקודה tree על תיקיות אלו, כאשר מימין לכל קובץ מופיע y אם
 הוא טוב או n אם הוא אינו טוב.

```
d1
|-- A y
|-- AA
|   |-- F1 n
|   `-- F2 y
|-- CC n
|-- DDD y
`-- EE
    `-- F1 n
```

```
d2
|-- A1
|   |-- F1 n
|   `-- F2 n
`-- XXX n
```

```
d3
|-- B
|   |-- CC y
|   `-- F2 n
|-- F1 y
`-- XXX n
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P8.1 d3 d1 d2
```

יתקבל הפלט:

```
d3 2
d1 3
d2 0
```

```
dirs=$*
for dir in $dirs
do
  counter=0
  for file in $(find $dir -type f)
  do
    exist=0
    NumOfLines=$(wc -l < $file)
    while read line
    do
      NumOfWords=$(echo "$line" | wc -w)
      if [ $NumOfWords -gt $NumOfLines ]
      then
        exist=0
        break
      fi
      if [ $NumOfWords -eq $NumOfLines ]
      then
        ((exist++))
      fi
    done<$file
    if [ $exist -gt 0 ]
    then
      ((counter++))
    fi
  done
  echo "$dir $counter"
done
```

2. שאלה זו הופיע במבחן מועד ב 2015

כתוב/כתבי תכנית ב- Bash (דהינו קובץ Script) בשם P8.2 שמקבלת כפרמטרים רשימת תיקיות (בהמשך נקרא לה רשימה 1), לאחר מכן המילה files ולאחר מכן רשימת שמות קבצים (בהמשך נקרא לה רשימה 2), ומדפיסה לפלט שורה אחת עבור כל שם קובץ ברשימה 2 שמכילה את שם הקובץ, לאחריו תו רווח אחד ולאחריו עבור כל תיקיה ברשימה 1, זוג מהצורה: שם תיקיה ומספר, כאשר המספר שווה לסכום של מספרי השורות של כל הקבצים שנמצאים בתיקיה ושם הוא כשם הקובץ.

בכל זוג שמודפס ישנו תו רווח אחד בין התיקיה למספר ובין הזוגות ישנו תו רווח אחד.

על סדר השורות בפלט להיות לפי סדר הקבצים ברשימה 2 ובתוך כל שורה על סדר הזוגות להיות לפי סדר התיקיות ברשימה 1.

לדוגמה, להלן מבנה תיקיות d1 d2 d3 כפי שמתקבל על ידי הפעלת הפקודה tree על תיקיות אלו, כאשר מימין לכל קובץ מופיע מספר שמציין את מספר השורות של הקובץ (המספר הזה הוא לא חלק של הפלט של הפקודה tree ומוצג כאן לצורך הדוגמה).

```
d1
|-- A 20
|-- AA
|   |-- F1 3
|   `-- F2 2
|-- CC 7
`-- EE
    `-- F1 5
```

```
d2
|-- A1
|   |-- F1 1
|   `-- F2 6
`-- XXX 4
```

```
d3
|-- B
|   |-- CC 3
|   `-- F2 4
|-- CC 2
`-- XXX 5
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P8.2 d3 d1 d2 files XXX CC F1
```

יתקבל הפלט:

```
XXX d3 5 d1 0 d2 4
CC d3 5 d1 7 d2 0
F1 d3 0 d1 8 d2 1
```

```
rm -f dirs
for x in $*
do
  if [ "$x" = "files" ]
  then
    shift
    break
  fi
  echo $x >> dirs
  shift
done
for fileName in $*
do
  echo -n "$fileName"
  for dir in $(echo $(cat dirs))
  do
    sum=0
    echo -n " $dir"
    for file in $(find $dir -name $fileName)
    do
      sum=$((sum+$(wc -l < $file)))
    done
    echo -n " $sum"
  done
  echo
done
```

3. שאלה זו הופיע במבחן מועד ב 2015

הגדרה: נגדיר שמטריצה ריבועית היא טובה אם יש לה שתי עמודות זהות.

כתוב/כתבי תכנית ב- `bash` בשם `P8.3` שמקבלת כפרמטרים רשימת שמות קבצים שמכילים מטריצות ריבועיות (לא בהכרח באותו מימד). התכנית מדפיסה שורה אחד עבור כל קובץ שמכילה את שם הקובץ, לאחריו תו רווח אחד ולאחריו אחת משתי האפשרויות הבאות בהתאם לכך אם הקובץ הוא טוב או לא:

- אם הקובץ הוא טוב התכנית מדפיסה `yes` ולאחר מכן רשימה של מספרי כל העמודות במטריצה שיש להן לפחות עמודה אחת (נוספת) זהה, (עם תו רווח אחד בין המספרים). על המספרים ברשימה זו להיות שונים זה מזה וממוינים בסדר מספרי עולה.
- אם הקובץ אינו טוב התכנית מדפיסה `no`.

על השורות בפלט להיות ממוינות לפי שמות הקבצים בסדר לקסיקוגרפי עולה.

לדוגמה, נניח שתוכן הקבצים `G1-G4` הוא:

G1	G2	G3	G4
10 20 10 1 20	1 2 3 4	80	1 2 2 2
30 40 30 1 40	1 2 3 4		2 3 3 3
40 20 40 1 20	1 2 3 4		6 4 4 4
50 30 50 1 30	2 2 2 2		7 5 5 5
60 40 60 1 40			

לאחר הפעלת התכנית על ידי הפקודה:

```
P8.3 G2 G4 G3 G1
```

מתקבל הפלט:

```
G1 yes 1 2 3 5
G2 no
G3 no
G4 yes 2 3 4
```

```
rm -f result
for file
do
  rm -f file1
  while read line #file1 is constructed from $file
  do
    # by removing extra spaces
    echo $line >> file1
  done<$file
  cols=""
  n=$(wc -l < file1)
  for i in $(seq $n)
  do
    for j in $(seq ${i+1} $n)
    do
      col_i=$(cut -d" " -f$i file1)
      col_j=$(cut -d" " -f$j file1)
      if [ "$col_i" == "$col_j" ]; then
        cols="$cols $i $j"
      fi
    done
  done
  if [ "$cols" != "" ]; then
    cols1=$(echo $cols | tr " " "\n" | sort -n -u)
    echo $file yes $cols1 >> result
  else
    echo $file no >> result
  fi
done
sort result
```