

יטודות מערכות פתוחות
פתרון תרגיל מס' 5

שימו לב: כל ההערות שבתחילת תרגילים 1-3 תקפות גם לתרגיל זה.

הערה: בכל פעם שכתוב בתרגיל, סדר לכסיקוגרפי עולה הכוונה לסדר שמתקבל מהפעלת הפקודה `sort` ללא פרמטרים.

1. כתוב/כתבי תכנית `Script` ב-`Bash` בשם `P5.1` שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

```
Enter 2 parameters in format file-string:
```

והתכנית מחכה לקלט מהמשתמש שמכיל שם קובץ ומחרוזת למשל נניח שהמשתמש מקליד: `F1-a1b2c5`

נניח למשל שתוכן הקובץ `F1` הוא:

```
aa12bb34c5c z23ab3c21b5a
123cba 87aab312
67 8288a1bcc5 112233445aabddeecc
```

בשלב הזה התכנית מדפיסה לפלט את כל המילים בקובץ `F1` שמכילות את כל התווים שבמחרוזת. כל מילה בפלט תופיע בשורה נפרדת וסדר המילים הוא לפי סדר לכסיקוגרפי עולה.

בדוגמה הנ"ל, יתקבל הפלט:

```
112233445aabddeecc
8288a1bcc5
aa12bb34c5c
z23ab3c21b5a
```

פתרון שאלה 1

```
echo -n "Enter 2 parameters in format file-string: "  
read cmd  
echo -n "" >|result1  
filename=$(echo $cmd | cut -d"-" -f1)  
str=$(echo $cmd | cut -d"-" -f2)  
strlen=$(echo $str | egrep -o . | sort -u | wc -l)  
words=$(cat $filename)  
for w in $words; do  
    found=$(echo $w | egrep -o [$str] | sort -u | wc -l)  
    if [ $found -eq $strlen ]; then  
        echo $w >> result1  
    fi  
done  
sort result1
```

להלן אותו פתרון בפורמט טקסט.

```
echo -n "Enter 2 parameters in format file-string: "  
read cmd  
echo -n "" >|result1  
filename=$(echo $cmd | cut -d"-" -f1)  
str=$(echo $cmd | cut -d"-" -f2)  
strlen=$(echo $str | egrep -o . | sort -u | wc -l)  
words=$(cat $filename)  
for w in $words; do  
    found=$(echo $w | egrep -o [$str] | sort -u | wc -l)  
    if [ $found -eq $strlen ]; then  
        echo $w >> result1  
    fi  
done  
sort result1
```

2. כתוב/כתבי תכנית Script ב-Bash בשם P5.2 שפועלת באופן

הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

```
Enter list of files in format file1-file2-file3...:
```

והתכנית מחכה לקלט מהמשתמש שמכיל רשימת שמות קבצים. למשל נניח שהמשתמש מקליד: A1-A3-A2-A4

בשלב הזה מופיעה ההודעה הבאה:

```
Enter 2 numbers in format num1%num2:
```

והתכנית מחכה לקלט מהמשתמש שמכיל שני מספרים. למשל נניח שהמשתמש מקליד: 2%4

בשלב הזה התכנית מדפיסה לפלט את רשימת כל המילים שמספר ההופעות שלהן בכל אחד מהקבצים גדול או שווה ל- num1 וקטן או שווה מ- num2. כל מילה תופיע בפלט בשורה נפרדת בתוספת מספר שמציין את סך כל ההופעות של המילה בכל הקבצים. על הפלט להיות ממוין לפי מספר ההופעות של המילים בכל הקבצים, בסדר מספרי עולה. ניתן להניח שאין שתי מילים עם אותו מספר הופעות בכל הקבצים.

לדוגמה נניח שתוכן הקובץ A1 הוא:

```
ab 12ab 1cd uu 88 ab 33 33 1 1
ab cd uu 88 88 33 33 33 cw ab
```

ונניח שתוכן הקובץ A2 הוא:

```
5 ab 44 1cd 44 uu ab 1 1 1
ab 1cd 33 33 33 uu 88 88 cw 1
```

ונניח שתוכן הקובץ A3 הוא:

```
ab 44 88 88 1cd 33 33 44 uu ab
zx 1cd 33 44 uu 88 1 1 1 44
```

ונניח שתוכן הקובץ A4 הוא:

```
ab 88 12ab 1cd uu ab 33 ab 1 1 ab
zx 1cd uu 88 33 44 44 33
```

בדוגמה הנ"ל יתקבל הפלט:

```
uu 8
88 10
1 11
ab 13
```

פתרון שאלה 2

```
1 echo -n "Enter list of files in format file1-file2-file3...: "  
2 read a  
3 echo -n "Enter 2 numbers in format num1%num2: "  
4 read b  
5 files=$(echo $a | tr "-" " ")  
6 num1=$(echo $b | cut -d"%" -f1)  
7 num2=$(echo $b | cut -d"%" -f2)  
8 echo $(cat $files) | tr " " "\n" | sort|uniq -c >| all_files  
9 num_files=$(echo $files | wc -w)  
10 >| final  
11 while read line  
12 do  
13     num=$(echo $line | cut -d" " -f1)  
14     word=$(echo $line | cut -d" " -f2)  
15     counter=0  
16     for file in $files; do  
17         c=$(echo $(cat $file) | tr " " "\n" | egrep -c ^$word$)  
18         if [ $c -ge $num1 ] && [ $c -le $num2 ]; then  
19             ((counter++))  
20         fi  
21     done  
22     if [ $counter -eq $num_files ]; then  
23         echo $num" "$word" "$num >> final  
24     fi  
25 done <all_files  
26 sort -n final | cut -d" " -f2-  
27
```

להלן אותו פתרון בפורמט טקסט:

```
echo -n "Enter list of files in format file1-file2-file3...: "  
read a  
echo -n "Enter 2 numbers in format num1%num2: "  
read b  
files=$(echo $a | tr "-" " ")  
num1=$(echo $b | cut -d"%" -f1)  
num2=$(echo $b | cut -d"%" -f2)  
echo $(cat $files | tr " " "\n" | sort|uniq -c >| all_files  
num_files=$(echo $files | wc -w)  
>| final  
while read line  
do  
num=$(echo $line | cut -d" " -f1)  
word=$(echo $line | cut -d" " -f2)  
counter=0  
for file in $files; do  
c=$(echo $(cat $file) | tr " " "\n" | egrep -c ^$word$)  
if [ $c -ge $num1 ] && [ $c -le $num2 ]; then  
(counter++)  
fi  
done  
if [ $counter -eq $num_files ]; then  
echo $num" "$word" "$num >> final  
fi  
done <all_files  
sort -n final | cut -d" " -f2-
```

3. כתוב/כתבי תכנית Script ב-Bash בשם P5.3 שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

Enter file name + numbers in format file-num1%num2%num3...:

והתכנית מחכה לקלט מהמשתמש שמכיל שם קובץ ורשימת מספרים. נניח למשל שהמשתמש מקליד: G1-3%5%1%8%3

נניח למשל שתוכן הקובץ G1 הוא:

```
11 aa cccd 8  
abc gg zz aa d  
1 2 3 4 5 6 7  
zz yy zz 4  
11
```

בשלב הזה התכנית מדפיסה לפלט שורה אחת עבור כל מספר i ברשימת המספרים שמכילה את כל המילים שמופיעות כמילה i בקובץ באיזושהי שורה. על סדר המילים בשורה להיות לפי סדר לכסיקוגרפי עולה, ואין שתי מילים זהות באותה שורה. על פורמט הפלט להיות כמו בדוגמה.

עבור הקלט שבדוגמה הנ"ל יתקבל הפלט הבא:

```
3: 3 cccd zz
```

5: 5 d
1: 1 11 abc zz
8:
3: 3 cccd zz

שימו לב שבפלט הנ"ל בשורה של המספר 8 אין תו רווח מימין לנקודותיים, (דהינו בשורה הזו הנקודותיים הם סוף השורה ומיד לאחר מכן יש תו קפיצת שורה כפי שקיים בסוף כל שורה).

פתרון שאלה 3

```
1 echo -n "Enter file name + numbers in format file-num1%num2%num3...: "  
2 read a  
3 b=$(echo $a | tr "-" " " | tr "%" " ")  
4 file_name=$(echo $b | cut -d" " -f1)  
5 numbers=$(echo $b | cut -d" " -f2-)  
6 >| file  
7 while read line; do  
8     echo $line >> file  
9 done <$file_name  
10  
11 for num in $numbers; do  
12     if [ $num -eq 1 ]; then  
13         str="1: "$(cut -d" " -f1 file | sort -u)  
14         echo $str  
15     else  
16         str=$num": "$(cut -s -d" " -f$num file | sort -u)  
17         echo $str  
18     fi  
19 done  
20
```

להלן אותו פתרון בפורמט טקסט:

```
echo -n "Enter file name + numbers in format file-num1%num2%num3...: "  
read a  
b=$(echo $a | tr "-" " " | tr "%" " ")  
file_name=$(echo $b | cut -d" " -f1)  
numbers=$(echo $b | cut -d" " -f2-)  
>| file  
while read line; do  
    echo $line >> file  
done <$file_name  
  
for num in $numbers; do  
    if [ $num -eq 1 ]; then  
        str="1: "$(cut -d" " -f1 file | sort -u)  
        echo $str  
    else  
        str=$num": "$(cut -s -d" " -f$num file | sort -u)  
        echo $str  
    fi  
done
```

```
    str=$num": "$(cut -s -d" " -f$num file | sort -u)
    echo $str
fi
done
```


4. כתוב/כתבי תכנית Script ב-Bash בשם P5.4 שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

Enter formula:

והתכנית מחכה לקלט מהמשתמש שמכיל ביטוי אריתמטי שמורכב ממספרים ופעולות מהסוג * % + - /

נניח למשל שהמשתמש מקליד:

6+5%3-140/18

בשלב הזה יש שתי אפשרויות, אם הביטוי שהמשתמש הקליד חוקי התכנית מדפיסה את המספרים שמופיעים בביטוי לפי סדר הופעתם בביטוי. אם הביטוי אינו חוקי התכנית מדפיסה לפלט את ההודעה הבאה:

Illegal arithmetic expression

למשל עבור הקלט הנ"ל התכנית תדפיס לפלט:

6 5 3 140 18

ביטוי נחשב לא חוקי אם יש שני אופרטורים רצופים ללא מספר ביניהם. או שישנו אופרטור מסוים שאין מספרים משני הצדדים שלו. ניתן להניח שהקלט יכיל רק מספרים ואופרטורים מהסוג * % + - /

דוגמה לביטויים לא חוקיים:

+5

-5

60+70%-4

80+-50

5++40

60*

עבור כל אחד מהביטויים הלא חוקיים הנ"ל התכנית תדפיס את הודעת השגיאה הנ"ל.

פתרון שאלה 4

```
1 errmsg="Illegal arithmetic expression"
2 ops="-+/*%"
3 echo -n "Enter formula: "
4 read cmd
5 check=$(echo $cmd | egrep -c "[$ops][$ops]|^[$ops]|[$ops]$")
6 if [ $check -eq 0 ]; then
7     result=$(echo $cmd | tr "$ops" " ")
8     echo $result
9 else
10    echo $errmsg
11 fi
```

להלן אותו פתרון בפורמט טקסט:

```
errmsg="Illegal arithmetic expression"
ops="-+/*%"
echo -n "Enter formula: "
read cmd
check=$(echo $cmd | egrep -c "[$ops][$ops]|^[$ops]|[$ops]$")
if [ $check -eq 0 ]; then
    result=$(echo $cmd | tr "$ops" " ")
    echo $result
else
    echo $errmsg
fi
```

