

יסודות מערכות פתוחות
פתרון תרגיל מס' 7

שימו לב: כל ההערות שבתחילת תרגילים 6-1 תקפות גם לתרגיל זה.

הערה 1: החל מתרגיל זה והלאה כשכתוב למשל שתוכנית P1 מקבלת כפרמטר שם קובץ (או מחרוזת וכדומה) הכוונה שהפרמטר מופיע כבר בקריאה לתוכנית. לדוגמה הקריאה לתוכנית תהיה מהצורה: P1 F1. (זה בניגוד לתרגילים הקודמים שבהם הקריאה לתוכנית הייתה מהצורה P1 ורק לאחר מכן המשתמש התבקש להקליד מחרוזת שמכילה את שם הקובץ F1).

הערה 2: החל מתרגיל זה והלאה, בכל פעם שכתוב מילה הכוונה לרצף כלשהו של תווים ללא תווי רווח או סוף שורה. לדוגמה המחרוזת abc@#&cd נחשבת למילה אחת.

1. כתוב תוכנית Script ב-Bash בשם P7.1 שמקבלת כפרמטר שם קובץ ומדפיסה לפלט את כל השורות בקובץ שמקיימות את התנאי הבא:

יש במילה הראשונה בשורה איזושהי מחרוזת שמופיעה פעמים (באיזשהו מקום במילה) ואותה מחרוזת מופיעה במילה האחרונה בשורה לפחות פעמים ברצף.

לדוגמה, נניח שתוכן הקובץ F1 הוא:

```
12ab@5dab @3a@3a@3a@3a 1abab@5d
cdcd 77cdcd88
cd8cd 77cd8cd
a11bbb1111 88 ab1121111
abcdeabc8abcde abcdeabcde
abcdeabc8abcde abcdeabcd
12341234 123412341234
```

לאחר הפעלת התוכנית על ידי הפקודה P7.1 F1 מתקבל הפלט:

```
12ab@5dab @3a@3a@3a@3a 1abab@5d
cdcd 77cdcd88
a11bbb1111 88 ab1121111
abcdeabc8abcde abcdeabcde
12341234 123412341234
```

פתרון שאלה 7.1

```
1 counter=0
2 while read line; do
3     ((counter++))
4     firstWord=$(echo $line | tr " " "\n" | head -1)
5     lastWord=$(echo $line | tr " " "\n" | tail -1)
6     f_l="$firstWord $lastWord"
7     check=$(echo $f_l | egrep -c "^[^ ]*([ ]+)[^ ]*\1[^ ]*[ ][^ ]*\1\1[^ ]*$")
8     if [ $check -eq 0 ];then
9         continue
10    fi
11    head -$counter $1 | tail -1
12 done<$1
13
```

להלן אותו פתרון בפורמט טקסט.

```
counter=0
while read line; do
    ((counter++))
    firstWord=$(echo $line | tr " " "\n" | head -1)
    lastWord=$(echo $line | tr " " "\n" | tail -1)
    f_l="$firstWord $lastWord"
    check=$(echo $f_l | egrep -c "^[^ ]*([ ]+)[^ ]*\1[^ ]*[ ][^ ]*\1\1[^ ]*$")
    if [ $check -eq 0 ];then
        continue
    fi
    head -$counter $1 | tail -1
done<$1
```

.2

הגדרה 1: נגדיר ערך של תו בהתאם למספר i באופן הבא:
אם התו אינו אות אנגלית קטנה הערך הוא 0
אם התו הוא a הערך הוא i
אם התו הוא b הערך הוא $2i$
...
אם התו הוא z הערך הוא $26i$

הגדרה 2: נגדיר ערך של מילה בהתאם למספר i כסכום של הערכים של התווים שבמילה בהתאם למספר i .

לדוגמה הערך של המילה `abAz@1` בהתאם למספר 2 הוא:
 $2+4+0+52+0+0=58$

כתוב תוכנית `Script` ב-`Bash` בשם `p7.2` שמקבלת כפרמטרים מספר i ומילה.

התכנית מדפיסה לפלט שורה אחת שמכילה את הערך של המילה בהתאם למספר i .

לדוגמה, לאחר הפעלת התכנית על ידי הפקודה `abAz@1 2 p7.2` מתקבל הפלט:

58

לאחר הפעלת התכנית על ידי הפקודה `aabb##cc 2 p7.2` מתקבל הפלט:
24

הדרכה

הביטוי `{a..z}` מתפרש על ידי הסורק כסדרת התווים `a b c ...z`

לדוגמה אם רוצים להדפיס את התווים `a-z` כאשר כל תו מופיע בשורה נפרדת אפשר להשתמש בקטע הקוד הבא:

```
for dig in {a..z}
do
    echo $dig
done
```

פתרון שאלה 7.2

```
1 sum=0
2 for char in $(echo $2 | grep -o .); do
3     if [ $(echo $char | egrep -c '[a-z]') -eq 0 ];then
4         continue
5     fi
6     val=1
7     for letter in {a..z}; do
8         if [ $char = $letter ]; then
9             ((sum += val * $1))
10            break
11        fi
12        ((val++))
13    done
14 done
15 echo $sum
```

להלן אותו פתרון בפורמט טקסט.

```
sum=0
for char in $(echo $2 | grep -o .); do
    if [ $(echo $char | egrep -c '[a-z]') -eq 0 ];then
        continue
    fi
    val=1
    for letter in {a..z}; do
        if [ $char = $letter ]; then
            ((sum += val * $1))
            break
        fi
        ((val++))
    done
done
echo $sum
```

.3

כתוב תוכנית Script ב-Bash בשם P7.3 שמקבלת כפרמטרים מספר i ושם קובץ.

התכנית מדפיסה לפלט שורה אחת עבור כל ערך j של מילה בקובץ בהתאם למספר i שמכילה את המספר j ולאחריו רשימת כל המילים בקובץ שהערך שלהם בהתאם למספר i שווה ל-j. הפורמט של הפלט הוא כפי שמתואר בדוגמה הבאה.

על שורות הפלט להיות ממוינות לפי סדר מספרי עולה של הערכים שבתחילת כל שורה. ובתוך כל השורה על המילים להיות ממוינות לפי סדר לכסיקוגרפי עולה

לדוגמה נניח שתוכן הקובץ F3 הוא:

```
abzcaaa 11@bd aaaaaa bbb
ggg77 fzabc czdaa
klg77#a dddz 888xeebb yabc
iz 7zbbfb
```

לאחר הפעלת התכנית על ידי הפקודה F3 2 P7.3 מתקבל הפלט:

```
12: 11@bd aaaaaa bbb
42: ggg77
62: klg77#a yabc
70: abzcaaa czdaa iz
76: 7zbbfb 888xeebb dddz fzabc
```

הדרכה

לפתרון השאלה ניתן להניח שבתיקיה בה מופעלת התכנית P7.3 נמצאת גם התכנית P7.2 ולכן ניתן לקרוא לה כדי לחשב את הערך של מילה בהתאם למספר i.

פתרון שאלה 7.3

```
i=$1
file=$2
>|tmp
for word in $(cat $file); do
    value=$(P7.2 $i $word)
    echo $value $word >>tmp
done

values=$(cut -d" " -f1 tmp | sort -nu)
for v in $values; do
    words=$(egrep "^$v " tmp | cut -d" " -f2 | sort)
    echo $v": "$words
done
```

להלן אותו פתרון בפורמט טקסט.

```
i=$1
file=$2
>|tmp
for word in $(cat $file); do
    value=$(P7.2 $i $word)
    echo $value $word >>tmp
done

values=$(cut -d" " -f1 tmp | sort -nu)
for v in $values; do
    words=$(egrep "^$v " tmp | cut -d" " -f2 | sort)
    echo $v": "$words
done
```

4. שאלה זו הופיעה במבחן מועד ג בשנת 2016.

כתוב/כתבי תכנית ב-Bash (דהינו קובץ Script) בשם P7.4 שמקבלת כפרמטרים רשימת מחרוזות (בהמשך נקרא לה רשימת המחרוזות), לאחריה המילה dirs ולאחריה רשימת תיקיות בהמשך נקרא לה רשימת התיקיות), ומדפיסה לפלט את השמות של כל הקבצים שמקימים את התנאי שמוסבר להלן. לדוגמה, שם הקובץ F1 מקיים את התנאי אם בכל אחת מהתיקיות שברשימת התיקיות יש קובץ ששמו F1 שמכיל את כל המחרוזות שברשימת המחרוזות.

התכנית תדפיס את שמות כל הקבצים שמקימים את התנאי הנ"ל, כאשר כל שם קובץ מופיע בשורה נפרדת ושמות הקבצים ממוינים בסדר לכסיקוגרפי עולה.

שימו לב שהכוונה בתנאי שקובץ מכיל מחרוזת, היא שהמחרוזת נמצאת באיזשהו מקום בקובץ ויתכן גם בתוך מחרוזת אחרת. לדוגמא, אם בקובץ F1 יש שורה שמכילה את המחרוזת abcdef אז הקובץ F1 מכיל את המחרוזת abcdef וגם את כל התת מחרוזות שלה כמו, ab cd abc וכו'.

ראה/י דוגמה בעמוד הבא

לדוגמה, להלן תאור של מבנה תיקיות לאחר ביצוע הפקודה tree כאשר ליד כל קובץ מופיע y אם הקובץ מכיל את כל המחרוזות שברשימת המחרוזות: abc def gh או n, אחרת, (התוספת הזו היא רק לצורך ההסבר של הדוגמה ואינה מופיעה כפלט של הפקודה tree).

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

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

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

לאחר הפעלת התכנית ע"י הפקודה:

```
P7.4 abc def gh dirs d1 d2 d3
```

יתקבל הפלט:

```
CC
F1
F2
```

הסבר לפלט: השם CC מופיע בפלט כי בכל אחת מהתיקיות ישנו קובץ בשם CC שמכיל את כל המחרוזות abc def gh.

הדרכה כדי שתוכלו לבדוק את התכנית שלכם תעתיקו את התיקיה ~basicsys/win17/ex7/t

לחשבון שלכם ואז תריצו את הפקודה שבדוגמה הנ"ל ותבדקו שהפלט יוצא כמו בדוגמה.


```

strings=""
while [ $1 != "dirs" ]; do
    strings="$strings $1"
    shift
done
shift
dirs="$@"
num_strings=$(echo $strings | wc -w)
num_dirs=$(echo $dirs | wc -w)
find $dirs -type f >| tmp
>|files
for file in $(cat tmp); do
    echo $file | tr "/" "\n" | tail -1 >> files
done

for file in $(sort -u files); do
    ok_dir=0
    for dir in $dirs; do
        find $dir -type f -name $file >| tmp1
        for f in $(cat tmp1); do
            ok_strings=1
            for str in $strings; do
                if [ $(egrep -c "$str" $f) -eq 0 ];then
                    ok_strings=0;
                    break
                fi
            done
            if [ $ok_strings -eq 1 ]; then
                ((ok_dir++))
                break
            fi
        done
    done
    if [ $ok_dir -eq $num_dirs ]; then
        echo $file
    fi
done

```

להלן אותו פתרון בפורמט טקסט.

```
strings=""
while [ $1 != "dirs" ]; do
    strings="$strings $1"
    shift
done
shift
dirs="$@"
num_strings=$(echo $strings | wc -w)
num_dirs=$(echo $dirs | wc -w)
find $dirs -type f >| tmp
>|files
for file in $(cat tmp); do
    echo $file | tr "/" "\n" | tail -1 >> files
done
for file in $(sort -u files); do
    ok_dir=0
    for dir in $dirs; do
        find $dir -type f -name $file >| tmp1
        for f in $(cat tmp1); do
            ok_strings=1
            for str in $strings; do
                if [ $(egrep -c "$str" $f) -eq 0 ];then
                    ok_strings=0;
                    break
                fi
            done
            if [ $ok_strings -eq 1 ]; then
                ((ok_dir++))
                break
            fi
        done
    done
    if [ $ok_dir -eq $num_dirs ]; then
        echo $file
    fi
done
```