

יסודות מערכות פתוחות  
פתרון תרגיל מס' 6

שימו לב: כל ההערות שבתחילת תרגילים 5-1 תקפות גם לתרגיל זה.

1. כתוב תוכנית Script ב-Bash בשם P6.1 שמקבלת כפרמטרים שני מספרים שלמים גדולים מ-1 (בהמשך נקרא להם i ו-j) ולאחריהם רשימת מילים. התוכנית מדפיסה לפלט את המילים ברשימה שניתנות לחלוקה ל - x מילים זהות כאשר x הוא מספר שלם כלשהו בין i ל-j (כולל i ו-j).  
על כל מילה להיות מודפסת בשורה נפרדת וסדר המילים בפלט צריך להיות לפי סדר הופעתן ברשימת הפרמטרים. ניתן להניח שהפרמטר הראשון i קטן מהפרמטר השני j.

דוגמאות:

לאחר הפעלת התוכנית ע"י הפקודה:

```
P6.1 2 4 123123123 aaaaa abababab aaab baaa abcde1abcde1  
aaaaaa 1212121212
```

יתקבל הפלט הבא:

```
123123123  
abababab  
abcde1abcde1  
aaaaaa
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P6.1 3 5 123123123 aaaaa abababab a1a1a1a1a1a1 a1a1a  
a1a1a1a1a1a1a bbbbbbb
```

יתקבל הפלט הבא:

```
123123123  
aaaaa  
abababab  
a1a1a1a1a1a1  
a1a1a1a1a1a1a
```

## פתרון שאלה 6.1:

```
i=$1; j=$2
shift
shift
i=${i-1}
j=${j-1}
for str in $@
do
    echo "$str" | egrep "^(.+)\1{$i,$j}$"
done
```

ובפורמט טקסט:

```
i=$1; j=$2
shift
shift
i=${i-1}
j=${j-1}
for str in $@
do
    echo "$str" | egrep "^(.+)\1{$i,$j}$"
done
```

2. כתוב תוכנית Script ב- Bash בשם P6.2 שמקבלת כפרמטרים מספר (בהמשך נקרא לו i) לאחר מכן שם קובץ (בהמשך נקרא לו קובץ 1) ולאחר מכן רשימת תווים (אורך רשימת התווים אינו מוגבל). התוכנית ומדפיסה לפלט את כל המילים בקובץ 1 שמספר ההופעות בהן של כל אחד מהתווים ברשימה שווה בדיוק ל- i. על כל מילה להיות מודפסת בשורה נפרדת (פעם אחת בדיוק) וסדר המילים בפלט צריך להיות לפי סדר לכסיקוגרפי עולה (בהתאם לפקודה sort ללא אופציות).

דוגמאות:

נניח שתוכן הקובץ F1 הוא:

```
wab 12aaabbb abababx ab ttttt baaabb zabcabc
baab baaabb cbaab ab ccabab zzz
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P6.2 3 F1 a b
```

יתקבל הפלט הבא:

```
12aaabbb
abababx
baaabb
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P6.2 2 F1 a b c
```

יתקבל הפלט הבא:

```
ccabab
zabcabc
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P6.2 0 F1 a b c
```

יתקבל הפלט הבא:

```
ttttt
zzz
```

## פתרון שאלה 6.2:

```
echo -n "">|tmp
len=$1; file=$2; shift; shift
for x in $(cat $file)
do
    goodword=1
    for char in @$@
    do
        if [ $(echo $x | egrep -o $char | wc -l) -ne $len ]
        then
            goodword=0
            break
        fi
    done
    if [ $goodword -eq 1 ]
    then
        echo $x>>tmp
    fi
done
sort -u tmp
```

ובפורמט טקסט:

```
echo -n "">|tmp
len=$1; file=$2; shift; shift
for x in $(cat $file)
do
    goodword=1
    for char in @$@
    do
        if [ $(echo $x | egrep -o $char | wc -l) -ne $len ]
        then
            goodword=0
            break
        fi
    done
    if [ $goodword -eq 1 ]
    then
        echo $x>>tmp
    fi
done
sort -u tmp
```

### 3. שאלה זו הופיעה במבחן מועד א בשנת 2017.

נגדיר שקובץ הוא מטריצה ריבועית אם בכל שורה בקובץ יש אותו מספר מילים ומספר השורות בקובץ שווה למספר המילים בשורה.

לדוגמה, הקבצים H1 H2 הם מטריצות ריבועיות והקבצים G1 G2 אינם מטריצות ריבועיות:

H1	H2	G1	G2
10 20 ab -30 cd 10 a b c	10	10 20	ab cd jj 12 34

כתוב/כתבי תכנית ב- Bash (דהינו קובץ Script) בשם P6.3 שמקבלת כפרמטרים רשימת תיקיות ומדפיסה לפלט את שמות כל הקבצים שמכילים מטריצות ריבועיות ונמצאים באחת מהתיקיות (בעומק כלשהו). כל שם קובץ יופיע בשורה נפרדת ויתאר את המסלול אל הקובץ החל מהתיקה הנוכחית בה מריצים את התכנית P6.3. על סדר הקבצים בפלט להיות לפי השמות הסופיים של הקבצים בסדר לכסיקוגרפי עולה. קבצים עם שם סופי זהה ימוינו ביניהם לפי סדר לכסיקוגרפי של המסלול אליהם.

רמז: כדי ליצור את המיון הנ"ל אפשר להוסיף לכל מסלול שמתאר קובץ את שם הקובץ הסופי לתחילת המסלול ואז לבצע את המיון. לאחר המיון יש לחתוך את השם שהוסף לתחילת המסלול של כל קובץ.

לדוגמה, נניח שקיבלנו את הקבצים הבאים שמתוארים על ידי המסלולים אליהם אבל עדין לא מסודרים:

d2/A  
d1/d2/A  
d2/d4/B  
d1/d2/CC  
d2/d4/d5/A

בשלב ראשון נוסיף את השם הסופי לכל מסלול ונקבל:

A/d2/A  
A/d1/d2/A  
B/d2/d4/B  
CC/d1/d2/CC  
A/d2/d4/d5/A

עכשיו נפעיל את הפקודה sort ונקבל:

A/d1/d2/A  
A/d2/A  
A/d2/d4/d5/A  
B/d2/d4/B  
CC/d1/d2/CC

עכשיו נחתוך את השם שהוספנו לתחילת כל מסלול ונקבל  
את התוצאה הסופית המבוקשת:

d1/d2/A  
d2/A  
d2/d4/d5/A  
d2/d4/B  
d1/d2/CC

ראה/י דוגמה בעמוד הבא.

לדוגמה, להלן מבנה תיקיות d1 d2 שנמצאות בתיקיה הנוכחית בה מריצים את התכנית P6.3. המבנה מתואר כפי שמתקבל על ידי הפעלת הפקודה tree על תיקיות אלה, כאשר מימין לכל קובץ מופיע y או n בהתאם לכך אם הקובץ מכיל מטריצה ריבועית או לא. (התוספת הזו היא רק לצורך ההסבר של הדוגמה ואינה מופיעה כפלט של הפקודה tree).

```
d1
|-- A  n
|-- d2
|   |-- A  y
|   |-- CC y
|   |-- F1 n
|   `-- F2 y
|-- CC  n
|-- F1  y
`-- d3
    `-- F1  n
```

```
d2
|-- A  y
|-- d4
|   |-- B  y
|   |-- d5
|   |   |-- A  y
|   |   `-- CC y
|   |-- F1  n
|   `-- F2  n
`-- CC  n
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P6.3 d1 d2
```

יתקבל הפלט:

```
d1/d2/A
d2/A
d2/d4/d5/A
d2/d4/B
d1/d2/CC
d2/d4/d5/CC
d1/F1
d1/d2/F2
```

כדי לבדוק שהתכנית שלכם עובדת נכון העתיקו את התיקיות d1 ו-d2

אל התיקיה שבה אתם מריצים את התכנית P6.3 על ידי הפקודה:

```
cp -r ~basicsys/win18/ex6/d[1-2] .
```

בשלב הזה תפעילו את התכנית P6.3 d1 d2 ותבדקו שהפלט הוא כפי שתואר בעמוד הקודם.

### פתרון שאלה 6.3:

```
function check_square_matrix {
  local file=$1 n=$(wc -l < $file) result=YES line
  while read line; do
    if [ $(echo $line | wc -w) -ne $n ]; then
      result=NO
      break
    fi
  done<$file
  echo $result
}
for file in $(find $@ -type f); do
  if [ $(check_square_matrix $file) = YES ]; then
    echo $file
  fi
done >| tmp
for file in $(cat tmp); do
  echo $(echo $file | tr "/" "\n" | tail -1)/$file
done >| tmp1
sort tmp1 | cut -d"/" -f2-
```

ובפורמט טקסט:

```
function check_square_matrix {
  local file=$1 n=$(wc -l < $file) result=YES line
  while read line; do
    if [ $(echo $line | wc -w) -ne $n ]; then
      result=NO
      break
    fi
  done<$file
  echo $result
}
for file in $(find $@ -type f); do
  if [ $(check_square_matrix $file) = YES ]; then
    echo $file
  fi
done >| tmp
for file in $(cat tmp); do
  echo $(echo $file | tr "/" "\n" | tail -1)/$file
done >| tmp1
sort tmp1 | cut -d"/" -f2-
```



4. כתוב תוכנית Script ב-Bash בשם P6.4 שמדפיסה את השעה ב-Sydney שבאוסטרליה על ידי לקיחת האינפורמציה מהאתר הבא:

```
https://www.worldtimeserver.com/
```

על הפלט להראות רק שעות ודקות בתוספת של AM או PM בהתאם לשעה ועל ההודעה להיות מודפסת לפי הפורמט שמתואר בדוגמה הבאה.

לדוגמה, בזמן הכנת תרגיל בית זה השעה בסידני הייתה 10:19:27 PM. במצב זה על התכנית להדפיס את ההודעה הבאה:

```
The time now at Sydney Australia is: 10:19 PM
```

**רמז:** כדי לבצע את התרגיל יש להכנס לאתר הנ"ל ולחפש בו Sydney לאחר החיפוש יש להעתיק את הקישור שאליו מגיעים ולהפעיל את curl על קישור זה כאשר הפלט של curl מוצב לקובץ זמני למשל tmp.

בשלב זה יש להסתכל על השעה שמראה האתר בסידני ולנסות לחפש איפה בקובץ tmp מופיעה השעה הזו. אפשר לעשות זאת בשיטה שהוצגה בהרצאה על ידי שימוש ב-egrep -o. לאחר מציאת המקום שבו נשמרת השעה והדקה (רמז: אפשר לחפש למשל 10:19), יש לכתוב את התכנית P6.4 שתיצור את קובץ tmp באופן הנ"ל, תשלוף ממנו את השעה הדקה והאם הזמן הוא AM או PM ותדפיס את ההודעה המתאימה.

## פתרון שאלה 6.4:

```
curl -s https://www.worldtimeserver.com/current_time_in_AU-NSW.aspx >| tmp
full_time=$(cat tmp | egrep -o "[0-9]?[0-9]:[0-9]{2}:[0-9]{2} (AM|PM)" | tail -1)
hhmm=$(echo $full_time | cut -d":" -f1-2)
ampm=$(echo $full_time | cut -d" " -f2)
echo "The time now at Sydney Australia is: $hhmm $ampm"
rm -f tmp
```

ובפורמט טקסט:

```
curl -s https://www.worldtimeserver.com/current_time_in_AU-NSW.aspx >| tmp
full_time=$(cat tmp | egrep -o "[0-9]?[0-9]:[0-9]{2}:[0-9]{2} (AM|PM)" | tail -1)
hhmm=$(echo $full_time | cut -d":" -f1-2)
ampm=$(echo $full_time | cut -d" " -f2)
echo "The time now at Sydney Australia is: $hhmm $ampm"
rm -f tmp
```