

20.2.2014

פתרון מבחן מועד א'
יסודות מערכות פתוחות
סמסטר חורף, תשע"ד

1. (15 נקודות)
חלק א (5 נקודות)

מה יתקבל בפלט (דהינו יוצג על המסך) לאחר הפעלת קטע הקוד הבא:

```
echo 1 2 3 4 >|F1
echo 4 5 6 7 >>F1
echo 8 9 10 11 >>F1
echo 12 13 14 15 >>F1

n=$(head -1 F1 | wc -w)

for i in $(seq $n)
do
  for j in $(seq $n)
  do
    if [ ${i+$j} -eq ${n+1} ]
    then
      head -$i F1 | tail -1 | cut -d" " -f$j
    fi
  done
done
```

פתרון שאלה 1 חלק א:

4
6
9
12

חלק ב (10 נקודות)

מה יתקבל בפלט (דהינו יוצג על המסך) לאחר הפעלת קטע הקוד הבא:

```
echo "output 1:"
echo "123 456 789" >| F1
echo "a b c d e" >>F1
echo "hello hi" >|F2
cat F1 F2 >|F3
mkdir d1
cp F1 F2 d1
mv F3 d1
cat F1 - d1/* < F2 | wc -l

echo "output 2:"
echo "abcdefabf" |tr ab cd | egrep -o "[cdf]" | wc
-l

echo "output 3:"
echo "abc shalom" >|F1
echo " abc shalom 4" >> F1
echo "abc shalom12" >> F1
echo "abc shalom123 shalom6" >> F1
echo "abcshalom22 456 shalom" >> F1
egrep "(^[ ]*)[ ]+[ ]+shalom($|..[ ])" F1

echo "output 4:"
echo abcababcaba >|F1
echo 123c123123123123123123c123123123123 >> F1
echo acaaaca >> F1
echo accacacaaccacaca >> F1
egrep "^((.+ )c\2{2,4})\1$" F1

echo "output 5:"
echo "abyzx12" >|F1
echo "cdefgh" >> F1
echo "xyzxyzxyz" >> F1
cut -c1,3,5,7 F1 | egrep -o . | tr "\n" "," | \
cut -d"," -f1,3,5,7 --output-delimiter=""
```

פתרון שאלה 1 חלק ב:

output 1:

9

output 2:

8

output 3:

abc shalom

abcshalom22 456 shalom

output 4:

123c123123123123123c123123123123

acaaacaa

output 5:

axcg

2. (10 נקודות)

נגדיר שמילה היא רצף של תווים ללא תווי רווח וסוף שורה. כתוב תוכנית Script ב-sed בשם P2 שמקבלת כפרמטר שם קובץ (בהמשך נקרא לו קובץ 1) ומדפיסה לפלט את השורות בקובץ שהמילה הראשונה שבהן שווה למילה האחרונה כאשר בשורות שיודפסו משוכפלות הספרות החל מהספרה הרביעית והלאה, כפי שמודגם בדוגמה שלהלן. (ניתן להניח שבכל שורה יש לפחות שתי מילים).

על המבנה של התוכנית P2 להיות כדלהלן:

שורה ראשונה כמו בכל קובץ script ב-bash מכילה:

```
#!/bin/bash
```

לאחר מכן שורה אחת או יותר של פקודות sed לדוגמה

```
sed s/dog/cat/ $1
```

(אין להשתמש בפקודות שאינן של sed ושאין מתחילות ב-sed). מותר לכתוב לקובץ ביניים כמו למשל:

```
sed s/dog/cat/ $1 >| tmp
```

מותר גם לקרוא מקובץ ביניים כמו למשל:

```
sed s/dog/cat/ < tmp
```

אסור להשתמש ב - pipeline זאת אומרת המבנה הבא אסור :
sed s/dog/cat \$1 | sed s/abc/def

לדוגמה, נניח שתוכן הקובץ F2 הוא:

```
abc2 1 def2 3 abc2
123 zy
cd eabc1d rq12345 cd
a a
ala a1
```

לאחר הפעלת התוכנית ע"י הפקודה:

P2 F2

יתקבל הפלט:

```
abc2 1 def2 33 abc22
cd eabc1d rq12334455 cd
a a
```

פתרון שאלה 2:

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

3. (20 נקודות)

חלק א (15 נקודות)

כתוב/כתבי תוכנית ב-awk בשם P3.1 שמקבלת כפרמטרים מחרוזת (בהמשך נקרא לה מחרוזת1) ושם קובץ שמכיל מספרים (בהמשך נקרא לו קובץ1). המבנה של מחרוזתו הוא מספרים (במספר זוגי) כאשר בין כל שני מספרים מפריד תו פסיק אחד בדיוק. כל זוג מספרים שבמחרוזתו מיצג מספר בקובץ1. לדוגמה זוג המספרים 2,3 מיצג את המספר בקובץ1 שנמצא בשורה 2 ובעמודה 3. (דהינו המספר השלישי בשורה השנייה בקובץ1). התוכנית P3.1 מדפיסה לפלט את סכום המספרים בקובץ1 שמיוצגים על ידי זוגות המספרים שבמחרוזתו, כפי שמודגם בדוגמה הבאה. יתכן שבמחרוזתו מופיע זוג מספרים שהמספר המתאים לו בקובץ1 אינו קיים, כמו למשל מופיע זוג המספרים 2,5 ובקובץ1 בשורה השנייה יש רק ארבעה מספרים. במקרה כזה אין להתחשב בזוג המספרים הזה בחישוב הסכום. במילים אחרות, בחישוב הסכום אפשר לקחת 0 עבור זוג מספרים שמיוצג איבר שלא קיים בקובץ1. אין להשתמש בפקודה system בתוכנית P3.1. (דהינו בפתרון חלק א אסור להשתמש בפקודה system).

לדוגמה, נניח שתוכן הקובץ F1 הוא:

```
12 9 22
888 99
6 8 20 30
```

לאחר הפעלת התוכנית P3.1 ע"י הפקודה:

```
P3.1 "1,2,1,3,2,5,3,4,1,2" F1
```

יתקבל הפלט:

70

הפלט הנ"ל התקבל כתוצאה מחישוב הסכום הבא: $9+22+0+30+9$

פתרון שאלה 3 חלק א:

```
#!/bin/awk -f
BEGIN { str=ARGV[1];
        delete ARGV[1];
        split(str,NUMS,",");
        sum=0;
      }
{
  for (i=1;i<=length(NUMS);i+=2) {
    if (NUMS[i]==NR) {
      if (NUMS[i+1]<=NF) {
        sum+=$NUMS[i+1]
      }
    }
  }
}
END { print sum }
```

שאלה 3 חלק ב (5 נקודות)

כתוב/כתבי תוכנית ב-awk בשם P3.2 שמקבלת כפרמטרים מחרוזת (בהמשך נקרא לה מחרוזת1) ורשימת שמות קבצים שמכילים מספרים (בהמשך נקרא לה רשימה1). המבנה של מחרוזת 1 הוא כפי שתואר בחלק א. התוכנית P3.2 מדפיסה לפלט שורה אחת עבור כל קובץ שברשימת הקבצים שמכילה את שם הקובץ, לאחריה תו רווח ולאחריה מספר שמציין את סכום המספרים בקובץ שמיוצגים על ידי זוגות המספרים שבמחרוזת1, כפי שתואר בחלק א. על סדר השורות בפלט להיות לפי שמות הקבצים בסדר לכסיקוגרפי עולה (בהתאם לפקודה sort ללא אופציות). בתוכנית P3.2 מותר להשתמש לכל היותר ב-4 פקודות system.

לדוגמה, נניח שתוכן הקובץ F1 הוא כפי שתואר בחלק א, ותוכן הקובץ A הוא:

```
1 4 7 6 5
100 200 7 9 10
```

ונניח שתוכן הקובץ B הוא:

```
1000
```

לאחר הפעלת התוכנית P3.2 ע"י הפקודה:

```
P3.2 "1,2,1,3,2,5,3,4,1,2" F1 A B
```

יתקבל הפלט:

```
A 70
B 0
F1 25
```

פתרון שאלה 3 חלק ב:

```
#!/bin/awk -f
BEGIN { l=length(ARGV);
        str=ARGV[1];
        delete ARGV[1];
        split(str,NUMS,"");
    }
(FNR==1) { A[FILENAME]=0 }
{
    for (i=1;i<=length(NUMS);i+=2) {
        if (NUMS[i]==FNR) {
            if (NUMS[i+1]<=NF) {
                A[FILENAME]+=$NUMS[i+1]
            }
        }
    }
}
END { system("echo -n "" >| tmp")
      for (i=2;i<l;i++) {
          print ARGV[i] " "A[ARGV[i]] > "tmp"
      }
      system("cat tmp|sort")
}
```

4. (30 נקודות)

הגדרה: נגדיר שהערך של התו a שווה ל-1 הערך של התו b שווה ל-2 וכן הלאה עד התו z שערכו שווה ל-26. נגדיר ערך של מילה שמורכבת מאותיות אנגליות קטנות בלבד כסכום הערכים של התווים שבמילה. לדוגמה ערך המילה aabbcc שווה ל-12 (שהתקבל מהסכום $1+1+2+2+3+3$).

חלק א (10 נקודות)

כתוב/כתבי תוכנית ב-bash בשם P4.1 שמקבלת כפרמטר מילה שמורכבת מאותיות אנגליות קטנות בלבד ומדפיסה לפלט את ערך המילה.

אין להשתמש ב-awk ו-sed בתוכנית P4.1.

לדוגמה, לאחר הפעלת התוכנית P4.1 על ידי הפקודה:

P4.1 abcda

יתקבל הפלט:

11

הפלט התקבל מחישוב הסכום: $1+2+3+4+1$

פתרון שאלה 4 חלק א:

```
function calc_char {
  i=1
  for x in {a..z}
  do
    if [ $x = $1 ]
    then
      echo $i
      break
    fi
    i=${i+1}
  done
}
sum=0
for y in $(echo $1 | egrep -o ".")
do
  sum=$((sum+$(calc_char $y)))
done
echo $sum
```

חלק ב (20 נקודות)

כתוב/כתבי תוכנית ב- bash בשם P4.2 שמקבלת כפרמטרים מחרוזת (בהמשך נקרא לה מחרוזת1) ורשימת שמות קבצים שמכילים מילים שמורכבות מאותיות אנגליות קטנות בלבד. המבנה של מחרוזת 1 הוא כפי שתואר בשאלה 3 חלק א. התוכנית P4.2 מדפיסה לפלט שורה אחת עבור כל קובץ שברשימת הקבצים שמכילה את שם הקובץ, לאחריו תו רווח ולאחריו מספר שמציין את סכום ערכי המילים בקובץ שמיוצגים על ידי זוגות המספרים שבמחרוזת1, (כפי שתואר בשאלה 3 חלק א). על שורות הפלט להיות ממוינות לפי הסכומים בסדר מספרי עולה.

אין להשתמש ב- awk ו- sed בתוכנית P4.2. (מותר להשתמש בתוכנית P4.1 שכתבת בחלק א).

לדוגמה, להלן שמות קבצים ותוכנם

A	B	C	K	E
ab cde	a b c	a eee	abcde	aa bb cc dd ee
ef gh	d e f g h	g hh		ff
	d a b			a b c

לאחר הפעלת התוכנית ע"י הפקודה:

```
P4.2 "1,2,1,3,2,5,3,4,1,2" A B C K E
```

יתקבל הפלט:

K 0
E 14
B 15
A 24
C 30

המספר 14 שמופיע בפלט עבור הקובץ E התקבל מהסכום של ערכי המילים: bb, cc, bb
ששווה ל- $2+2+3+3+2+2$
(המילים שמתאימות לזוגות 2, 5 ו- 3, 4 לא קיימות בקובץ E ולכן מחושבות כ- 0 בסכום).

פתרון שאלה 4 חלק ב:

```
function get_str {
    file1=$1
    row1=$2
    col1=$3
    result=""
    num_of_rows=$(cat $file1 | wc -l)
    if [ $row1 -le $num_of_rows ]
    then
        line=$(cat $file1 | head -$row1 | tail -1)
        num_of_columns=$(echo $line | wc -w)
        if [ $col1 -le $num_of_columns ]
        then
            result=$(echo $line | cut -d" " -f$col1)
        fi
    fi
    echo $result
}
echo $1 | tr "," "\n" >| nums
c=$(cat nums | wc -l)
shift
echo -n "" >| tmp
for file in "$@"
do
    i=1
    sum=0
    while [ $i -lt $c ]
    do
        row=$(cat nums | head -$i | tail -1)
        col=$(cat nums | head -[$i+1] | tail -1)
        str=$(get_str $file $row $col)
        echo row=$row col=$col file=$file str=$str
        if [ "$str" != "" ]
        then
            sum=$((sum + $(P4.1 $str)))
        fi
        i=$((i+2))
    done
    echo "$file $sum" >> tmp
done
cat tmp | sort -k 2n,2
```

5. (25 נקודות)

כתוב/כתבי תוכנית ב-Bash (דהינו קובץ Script) בשם P5 שמקבלת כפרמטרים רשימת תיקיות.

התוכנית מדפיסה לפלט שורה אחת עבור כל תיקיה שמכילה את שם התיקייה לאחריה תו רווח אחד, לאחריו מספר שמציין את מספר השמות השונים של קבצים בתיקייה (בעומק כלשהו), ולאחריו רשימת שמות הקבצים שמופיעים בתיקייה מספר גדול ביותר של פעמים (מופרדים על ידי תו רווח יחיד). על סדר השורות בפלט להיות לפי סדר לכסיקוגרפי עולה של שמות התיקיות, ובתוך כל שורה על הסדר של שמות הקבצים להיות לפי סדר לכסיקוגרפי עולה.

לדוגמה, להלן מבנה תיקיות d1 d2 d3 כפי שמתקבל על ידי הפעלת הפקודה tree על תיקיות אלו. בתאור המבנה של התיקיות יש להניח ששם שאין מתחתיו קבצים מתאר קובץ (ולא תיקיה).

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

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

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

לאחר הפעלת התוכנית ע"י הפקודה:

```
P5 d3 d2 d1
```

יתקבל הפלט:

```
d1 5 F1
d2 4 CC F1 F2 XXX
d3 4 CC F1
```

```

echo -n "" >| tmp
dirs=$(echo "$@" | tr " " "\n" | sort \
      | tr "\n" " ")
for dir in $dirs
do
  find $dir -type f >| tmp1
  echo -n "" >| tmp2
  for x in $(cat tmp1)
  do
    echo $x | tr "/" "\n" | tail -1 >> tmp2
  done
  num=$(cat tmp2 | sort -u | wc -l)
  max=$(cat tmp2 | sort | uniq -c | \
        sort -n | tail -1)
  max=$(echo $max | cut -d" " -f1)
  cat tmp2 | sort | uniq -c >| tmp3
  file_list=""
  while read y
  do
    num1=$(echo $y | cut -d" " -f1)
    file=$(echo $y | cut -d" " -f2)
    if [ $num1 -eq $max ]
    then
      file_list="$file_list $file"
    fi
  done<tmp3
  sorted=$(echo $file_list | tr " " "\n" | sort \
          | tr "\n" " ")
  echo $dir $num $sorted
done

```