

יסודות מערכות פתוחות
פתרון תרגיל מס' 11

1. הגדרה: נגדיר את המסגרת ה- i של מטריצה ריבועית A כמסגרת של המטריצה שהאיבר השמאלי העליון שלה הוא $A_{i,i}$

לדוגמה נסתכל על המטריצה הריבועית A הבאה:

```
10 11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27
40 41 42 43 44 45
46 47 48 49 50 51
52 53 54 55 56 57
```

המסגרת ה-1 של המטריצה A הנ"ל היא:

```
10 11 12 13 14 15
16                      21
22                      27
40                      45
46                      51
52 53 54 55 56 57
```

המסגרת ה-2 של המטריצה A הנ"ל היא:

```
17 18 19 20
23          26
41          44
47 48 49 50
```

המסגרת ה-3 של המטריצה A הנ"ל היא:

```
24 25
42 43
```

עבור $i \geq 4$ למטריצה A אין מסגרת i .

כתוב/כתבי תכנית ב-`awk` בשם `p11.1` שמקבלת כפרמטרים רשימה של זוגות כאשר כל זוג מתאר סדרת מספרים ושם קובץ שמכיל מטריצה

ריבועית, התכנית מדפיסה לפלט שורה אחת עבור כל זוג שמתארת את הזוג, את סדרת הסכומים המתאימים, ו YES/NO בהתאם לכך אם הסדרה היא סדרה חשבונית או לא. המבנה של הזוגות המועברים כפרמטרים לתכנית והמבנה של שורות הפלט הוא כפי שמתואר בדוגמה שבהמשך. על הסדר של שורות הפלט להיות ממוין לפי המספר האחרון בשורה בסדר מספרי עולה. (ניתן להניח שאין שתי שורות עם אותו מספר אחרון בשורה).

לדוגמה, נניח שתוכן המטריצה A הוא:

```

1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 7 5 5 5 1 1
1 1 1 1 2 1 1 1
1 1 1 3 4 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

```

נניח שתוכן המטריצה B הוא:

```

1 1 1 1 1 1
1 2 2 2 2 1
1 2 3 4 2 1
1 2 5 8 2 1
1 2 2 2 2 1
1 1 1 1 1 9

```

נניח שתוכן המטריצה C הוא:

```

1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 1
1 2 3 3 3 3 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 4 4 3 2 1
1 2 3 3 3 3 3 2 1
1 2 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1 1

```

לאחר הפעלת התכנית על ידי הפקודה:

P11.1 "4,2,3:A" "1,3,2:A" "3,2,1:B" "1,2:C" "5,1,4:C"

יתקבל הפלט:

```

A 1 3 2 28 30 20 NO
B 3 2 1 20 24 28 YES
A 4 2 3 10 20 30 YES
C 5 1 4 5 32 32 NO
C 1 2 32 48 YES

```

מותר להשתמש בכלל היותר 4 פקודות system לפתרון השאלה.

מומלץ להשתמש בפונקציות של awk לפתרון התרגיל.

הסבר למבנה הפרמטרים: הפרמטר:

"4,2,3:A"

מתאר זוג שמופרד על ידי נקודותיים. החלק השמאלי של הזוג הוא סדרת מספרים מופרדת על ידי פסיקים, 4,2,3 והחלק הימני של הזוג מכיל שם קובץ שמכיל מטריצה ריבועית, דהינו המטריצה A.

הסבר לפלט: בשורה הראשונה רשומה שם המטריצה A לאחריה סדרת המספרים 1 3 2 ולאחריהם הסכומים של האיברים הנמצאים במסגרות המתאימות למספרים. למשל, הסכום 28 מתאים לסכום האיברים הנמצאים במסגרת 1 ששווה ל- 28 כי במסגרת הזו יש 28 אחדים. הסכום 30 מתאים לסכום האיברים שנמצאים במסגרת 3 שמתקבל מהסכום הבא:

$$7+5+5+5+1+1+1+1+1+1+1=30$$

הסכום 20 מתאים לסכום האיברים שנמצאים במסגרת 2 (כי במסגרת זו ישנם 20 אחדים). מאחר והסדרה 20 30 28 אינה סדרה חשבונית מודפס NO בסוף השורה הראשונה.

באופן דומה בשורה השניה מודפסת סדרת הסכומים 20 24 28 שמתאימה למסגרות 1 2 3 של המטריצה B ומודפס YES כי זו סדרה חשבונית.

הסדר של שורות הפלט נקבע לפי המספר האחרון בכל שורה לכן זה לפי הסדר הבא:

20
28
30
32
48

```

#!/bin/awk -f
BEGIN {
  for (x=1;x<length(ARGV);x++) {
    param[x]=ARGV[x]
    file=get_file(param[x])
    ARGV[x]=file
  }
}
{
  for (j=1;j<=NF;j++) {
    A[FILENAME,FNR,j]=$j
  }
  B[FILENAME]=FNR
}
END {
  for (x=1;x<=length(param);x++) {
    file=get_file(param[x])
    nums=get_nums(param[x])
    sums=get_sums(file,nums)
    seq=check_seq(sums)
    split(sums,D," ")
    last_sum=D[length(D)]
    print last_sum,file,nums,sums,seq > "tmp"
  }
  system("sort -n tmp | cut -d\" \" -f2-")
}
function get_file(par) {
  split(par,C,":")
  return C[2]
}
function get_nums(par,i) {
  split(par,C,":")
  split(C[1],D,",")
  str=D[1]
  for (i=2;i<=length(D);i++){
    str=str" "D[i]
  }
  return str
}

```

```

function get_sums(file,nums) {
  split(nums,C," ")
  sums=misgeret(file,C[1])
  for (i=2;i<=length(C);i++){
    sums=sums" "misgeret(file,C[i])
  }
  return sums
}
function misgeret(file,num) {
  n=B[file]
  sum=0
  if (num > n-num+1) {return sum}
  sum+=sum_row(file,num,num,n-num+1)
  if (num == n-num+1) {return sum}
  sum+=sum_row(file,n-num+1,num,n-num+1)
  if (num == n-num) {return sum}
  sum+=sum_col(file,num,num+1,n-num)
  sum+=sum_col(file,n-num+1,num+1,n-num)
  return sum
}
function sum_row(file,i,j1,j2,j,s){
  s=0
  for (j=j1;j<=j2;j++){
    s+=A[file,i,j]
  }
  return s
}
function sum_col(file,j,i1,i2,i,s){
  s=0
  for (i=i1;i<=i2;i++){
    s+=A[file,i,j]
  }
  return s
}
function check_seq(sums,i){
  split(sums,D," ")
  if (length(D)<3) {return "YES"}
  d=D[2]-D[1]
  for (i=3;i<=length(D);i++){
    if ((D[i]-D[i-1])!=d) { return "NO" }
  }
  return "YES" }

```

.2

כתוב/כתבי תכנית סקריפט ב- `bash` בשם P11.2 שמקבלת כפרמטרים מחרוזת וביטוי רגולארי ומדפיסה לפלט את החלקים של המחרוזת שהיו מוצבים למערך, אם הינו מפעילים את הפקודה `split` של `awk` על המחרוזת והביטוי הרגולארי. על מבנה הפלט להיות כפי שמתואר בדוגמה שבהמשך. אין להשתמש ב- `awk` או ב- `sed` לפתרון השאלה.

לדוגמה, לאחר הפעלת התכנית על ידי הפקודה:

```
P11.2 "xyababcdabababefab" "(ab)+"
```

יתקבל הפלט:

```
part 1 is:xy
part 2 is:cd
part 3 is:ef
part 4 is:
```

שימו לב שבשורה האחרונה בפלט אין כלום אחרי הנקודותיים כי מימין ל- `ab` יש מחרוזת ריקה.

לאחר הפעלת התכנית על ידי הפקודה:

```
P11.2 "abbbbcababcdabbbef" "ab+"
```

יתקבל הפלט:

```
part 1 is:
part 2 is:c
part 3 is:
part 4 is:cd
part 5 is:ef
```

```
string=$1
string2=$1
regexp=$2
echo $string | egrep -o "$regexp" >| tmp
count=1
while read line
do
  len=${#line}
  for i in $(seq 0 ${#string}-1)
  do
    if [ "${string:$i:$len}" = "$line" ];then
      echo part $count is:${string:0:$i}
      len_string=${#string}
      len1=${i+$len}
      string=${string:len1}
      ((count++))
      break
    fi
  done
done<tmp
echo part $count is:$string
```

3.

כתוב/כתבי תכנית סקריפט ב- `bash` בשם P11.3 שמקבלת כפרמטרים רשימה של זוגות ולאחריה שם קובץ ומדפיסה לפלט את תוכן הקובץ שמתקבל לאחר ההחלפות שמתבצעות בהתאם לרשימת הזוגות כפי שמתואר בדוגמה שבהמשך.
אין להשתמש ב- `awk` או ב- `sed` לפתרון השאלה.

לדוגמה, נניח שתוכן הקובץ F1 הוא:

```
ab cd ef gh
abbccddeeff hh
xyz abc qrw ztu
I am learning bash
123456789
```

לאחר הפעלת התכנית על ידי הפקודה:

```
P11.3 "1,3,5:abc" "3,1,6:def" "1,2,4:gh" "4,5,5:abab" F1
```

יתקבל הפלט:

```
aghc ef gh
abbccddeeff hh
defc qrw ztu
I amabablearning bash
123456789
```

שימו לב שאין לשנות את תוכן הקובץ F1.

הסבר לפרמטרים לתכנית: לדוגמה, המשמעות של הזוג:

```
"1,3,5:abc"
```

היא שיש להחליף בשורה הראשונה בקובץ את המילה שמתחילה בתו מספר 3 ומסתיימת בתו מספר 5 במחרוזת `abc`. בדוגמה הנ"ל תוחלף המילה `cd` שנמצאת בשורה הראשונה בקובץ בין תו 3 לתו 5 (כולל 5) במחרוזת `abc`.

מספור התווים בשורה מתחיל מ-1. ניתן להניח שהפרמטרים שמועברים לתכנית הם כאלו שאם מבקשים מילה שנמצאת בין תו `i` לתו `j` בשורה `k` אז יש כזו מילה בשורה `k`. (דהינו יש לפחות `k` שורות ויש לפחות `j` תווים בשורה `k`).

יש לבצע את ההחלפות לפי הסדר משמאל לימין. שימו לב שבהתחלה התו השלישי בשורה הראשונה מוחלף ב- `a` ולאחר מכן הוא מוחלף ב- `h`.

הסבר לפלט:

לדוגמה הפרמטר

```
"4,5,5:abab"
```

גורם לכך שהתו החמישי בשורה 4 (שהוא רווח) מוחלף במחרוזת `abab`.


```

function replace {
  file=$2
  pair=$1
  str=$(echo $pair | cut -d":" -f2)
  list=$(echo $pair | cut -d":" -f1)
  line_num=$(echo $list| cut -d"," -f1)
  from=$(echo $list| cut -d"," -f2)
  to=$(echo $list| cut -d"," -f3)
  n=$(wc -l < $file)
  for i in $(seq $n)
  do
    line_i=$(cat $file | head -$i | tail -1)
    if [ $i -ne $line_num ]; then
      echo "$line_i"
      continue
    fi
    left=${line_i:0:${from-1}}
    right=${line_i:$to}
    echo $left$str$right
  done
}

file=$(echo @$|cut -d" " -f$#)
pairs=$(echo @$|cut -d" " -f1-[$#-1])
cp $file tmp
for pair in $pairs
do
  replace $pair tmp >| tmp1
  mv tmp1 tmp
done
cat tmp

```

4. שאלה זו הופיע במבחן מועד ב 2015 (החומר לפתרון שאלה זו נמצא בחומר ללימוד עצמי כפי שמופיע באתר הקורס).

כתוב תכנית Script ב-sed בשם P11.4 שמקבלת כפרמטר שם קובץ ומדפיסה לפלט את השורות בקובץ שמכילות באיזשהו מקום בשורה מילה שמורכבת מאותיות אנגליות קטנות בלבד, לאחריה תו רווח אחד או יותר ולאחריה מילה שמורכבת מספרות בלבד. בנוסף בשורה האחרונה של הפלט מודפס מספר שמציין את מספר השורות שהודפסו.

על המבנה של התכנית P11.4 להיות כפי שתואר בשאלה 9.3.

לדוגמה, נניח שתוכן הקובץ F2 הוא:

```
abc2 1 def2 3 abc2
dea 123 zy45
cdeabc1d rq12345 cd 23
  abcabc 456ab
xyz%&12
hyz dd AB 123
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P11.4 F2
```

יתקבל הפלט:

```
dea 123 zy45
cdeabc1d rq12345 cd 23
2
```

פתרון שאלה 4

```
sed -r '/(^|[ ]+)([a-z]+)[ ]+([0-9]+)([ ]|$)/!d' $1 >| tmp
sed '/.*!/!d' tmp
sed -n '$=' tmp
```