

יטודות מערכות פתוחות
פתרון תרגיל מס' 8

שימו לב: כל ההערות שבתחילת תרגילים 1-7 תקפות גם לתרגיל זה.
הערה: כל התכניות ב- awk צריכות להכיל `#!/bin/awk-f` בשורה הראשונה.

1. כתוב תוכנית Script ב- `sed` (במבנה כפי שמתואר בתחילת שאלה 7.2) בשם `P8.1` שמקבלת כפרמטרים מחרוזת וקובץ ומדפיסה לפלט את השורות בקובץ שאינן מכילות את המחרוזת, כאשר בשורות אלה מופיעה המילה האחרונה שלוש פעמים ברצף (עם תו רווח אחד בין המילים), כפי שמודגם בדוגמה שלהלן.

לדוגמה, נניח שתוכן הקובץ `F1` הוא:

```
Helloabcd hi 123
if a equals b
you
one abc two
three aggbc four
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P8.1 abc F1
```

יתקבל הפלט:

```
if a equals b b b
you you you
three aggbc four four four
```

8.1 שאלה

```
sed '/!$1'/d' $2 >| tmp
sed -r 's/([ ]+)([ ]+)([ ]*)$/\1\2 \2 \2\3/' tmp
```

ובפורמט טקסט:

```
sed '/!$1'/d' $2 >| tmp
sed -r 's/([ ]+)([ ]+)([ ]*)$/\1\2 \2 \2\3/' tmp
```

2. כתוב תוכנית Script ב-awk בשם P8.2 שמקבלת כפרמטר שם קובץ שמכיל מטריצה (לא בהכרח ריבועית) של מספרים (לא בהכרח שלמים), ומדפיסה לפלט שורה אחת שמכילה את סכום המספרים בכל עמודה במטריצה כאשר בין המספרים בשורה ישנו תו רווח אחד בדיוק. אין להניח שמספר המספרים בכל שורה זהה.

לדוגמה, נניח שתוכן הקובץ F2 הוא:

```
5.3    4.6    2.1
10 12.8    3.3  18.9
4 15.2 20.444
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P8.2 F2
```

יתקבל הפלט:

```
19.3 32.6 25.844 18.9
```

8.2 פתרון שאלה

```
#!/bin/awk -f
function sum_col(A,j,n,s,i){
    s=0
    for (i=1;i<=n;i++){
        s+=A[i,j]
    }
    return s
}
BEGIN {
    file=ARGV[1]
    i=1;cols=1
    while (getline <file==1){
        for (j=1;j<=NF;j++){
            A[i,j]=$j
        }
        if (NF > cols) cols=NF
        i++
    }
    n=i--
    result=""
    for (j=1;j<=cols;j++){
        result=result" "sum_col(A,j,n)
    }
    sub("^[ ]"," ",result)
    print result
}
```

```
#!/bin/awk -f
function sum_col(A,j,n,s,i){
  s=0
  for (i=1;i<=n;i++){
    s+=A[i,j]
  }
  return s
}
BEGIN {
  file=ARGV[1]
  i=1;cols=1
  while (getline <file==1){
    for (j=1;j<=NF;j++){
      A[i,j]=$j
    }
    if (NF > cols) cols=NF
    i++
  }
  n=i--
  result=""
  for (j=1;j<=cols;j++){
    result=result" "sum_col(A,j,n)
  }
  sub("^[ ]"," ",result)
  print result
}
```

3. כתוב/כתבי תוכנית ב- awk בשם P8.3 שמקבלת כפרמטר שם קובץ. התוכנית מדפיסה לפלט מספר שמציין את אורך המילה הארוכה ביותר בקובץ מבין המילים שאינן מכילות כלל אותיות אנגליות (קטנות או גדולות).

לדוגמה, נניח שתוכן הקובץ F3 הוא:

```
hello 12345678hi you %&@1234
if else 111111111111H hi
abc      1234
```

לאחר הפעלת התוכנית ע"י הפקודה P8.3 F3 יתקבל הפלט:

7

8.3 פתרון שאלה

```
#!/bin/awk -f
BEGIN {
    file=ARGV[1]
    i=1;
    while (getline <file==1){
        for (j=1;j<=NF;j++){
            A[i,j]=$j
        }
        B[i]=NF
        i++
    }
    n=i--
    max=0
    for (i=1;i<=n;i++){
        for (j=1;j<=B[i];j++){
            if (sub("[a-zA-Z]", "&", A[i, j])==0 && length(A[i, j])>max){
                max=length(A[i, j])
            }
        }
    }
    print max
}
```

```

#!/bin/awk -f
BEGIN {
    file=ARGV[1]
    i=1;
    while (getline <file==1){
        for (j=1;j<=NF;j++){
            A[i,j]=$j
        }
        B[i]=NF
        i++
    }
    n=i--
    max=0
    for (i=1;i<=n;i++){
        for (j=1;j<=B[i];j++){
            if (sub("[a-zA-Z]", "&", A[i, j]) == 0 && length(A[i, j]) > max) {
                max=length(A[i, j])
            }
        }
    }
    print max
}

```

4. שאלה זו הופיע במבחן מועד ב 2017

הגדרה: עבור מטריצה A (לא בהכרח ריבועית) נגדיר את הקילוף ה- j, i של המטריצה, כמטריצה שמתקבלת מהמטריצה A לאחר הוצאת שורה i ועמודה j.

לדוגמה, עבור המטריצה A הבאה:

```
10 20 30 40 50
1 2 3 4 5 6 7
60 70 80
8 9
```

הקילוף ה- 2,3 של המטריצה A הוא המטריצה הבאה:

```
10 20 40 50
60 70
8 9
```

כתוב/כתבי תכנית ב- bash בשם P8.4 (אין להשתמש בפקודות awk ו- sed בשאלה זו) שמקבלת כפרמטר שם קובץ שמכיל מטריצה של מספרים (בהמשך נקרא לה מטריצה A) ובודקת האם קיימים 2 מספרים j, i כך שהקילוף ה- j, i של A מכיל 2 עמודות שסכום המספרים שלהן שווה. במידה והתשובה היא כן התכנית YES במידה ולא התכנית מדפיסה NO.

לדוגמה, נניח שנתונה מטריצה B הבאה:

```
10 20 40 50
60 70 20 28
8 9 50 40
```

לאחר הפעלת התכנית על ידי הפקודה:

```
P8.4 B
```

יתקבל הפלט:

```
YES
```

כי לאחר הקילוף 1,2 של המטריצה B מתקבלת המטריצה הבאה:

```
60 20 28
8 50 40
```

ובמטריצה זו יש 2 עמודות שהסכום שלהן שווה, דהיינו עמודות
1 -1 3

דוגמה נוספת, נניח שנתונה המטריצה C הבאה:

```
1 2 3
4 5 6
7 8 9
```

לאחר הפעלת התכנית על ידי הפקודה:

P8.4 C

יתקבל הפלט:

NO

כי בכל קילוף i, j של המטריצה C אין שתי עמודות שסכומן שווה.

פתרון שאלה 8.4

```
function sum_col {
    local file=$1 col=$2 s a z
    s=0;
    ((col--))
    while read line;do
        a=($line)
        z=${a[@]:$col:1}
        ((s+=z))
    done<$file
    echo $s
}

function num_of_cols {
    local file=$1 a z num_cols
    num_cols=0
    while read line;do
        a=($line)
        z=${#a[@]}
        if [ $z -gt $num_cols ]; then
            num_cols=$z
        fi
    done <$file
    echo $num_cols
}
```

```

function is_sum_two_cols_equal {
    local file=$1 j1 j2 n result
    n=$(num_of_cols $file)
    result=NO
    for j1 in $(seq $n); do
        for j2 in $(seq $n); do
            if [ $j1 -ne $j2 -a \
                $(sum_col $file $j1) -eq $(sum_col $file $j2) ]; then
                result=YES; break
            fi
        done
        if [ $result = YES ]; then
            break
        fi
    done
    echo $result
}

function kiluf_i_j {
    local file=$1 i=$2 j=$3 cnt=0 z
    while read line;do
        a=($line); ((cnt++)); z=${#a[@]}
        if [ $i -eq $cnt ]; then
            continue
        fi
        if [ $z -eq 1 ]; then
            if [ $j -eq 1 ]; then
                continue
            fi
            echo ${a[1]}
            continue
        fi
        if [ $j -eq 1 ]; then
            echo ${a[@]:1}
            continue
        fi
        echo ${a[@]}|cut -d" " -f1,${j}-1,${j+1}-
    done <$file
}

```



```

file=$1
n_rows=$(wc -l <$file)
n_cols=$(num_of_cols $file)
flag=NO
for i in $(seq $n_rows); do
  for j in $(seq $n_cols); do
    kiluf_i_j $file $i $j >| tmp
    flag=$(is_sum_two_cols_equal tmp)
    if [ $flag = YES ]; then
      break
    fi
  done
  if [ $flag = YES ]; then
    break
  fi
done
echo $flag

```

:8.4 טקסט טקסט טקסט

```

function sum_col {
  local file=$1 col=$2 s a z
  s=0;
  ((col--))
  while read line;do
    a=$line
    z=${a[@]:$col:1}
    ((s+=z))
  done<$file
  echo $s
}
function num_of_cols {
  local file=$1 a z num_cols
  num_cols=0
  while read line;do
    a=$line
    z=${#a[@]}
    if [ $z -gt $num_cols ]; then
      num_cols=$z
    fi
  done <$file
  echo $num_cols
}

```

```

function is_sum_two_cols_equal {
    local file=$1 j1 j2 n result
    n=$(num_of_cols $file)
    result=NO
    for j1 in $(seq $n); do
        for j2 in $(seq $n); do
            if [ $j1 -ne $j2 -a \
                $(sum_col $file $j1) -eq $(sum_col $file $j2) ]; then
                result=YES; break
            fi
        done
    if [ $result = YES ]; then
        break
    fi
done
echo $result
}

```

```

function kiluf_i_j {
    local file=$1 i=$2 j=$3 cnt=0 z
    while read line;do
        a=($line); ((cnt++)); z=${#a[@]}
        if [ $i -eq $cnt ]; then
            continue
        fi
        if [ $z -eq 1 ]; then
            if [ $j -eq 1 ]; then
                continue
            fi
            echo ${a[1]}
            continue
        fi
        if [ $j -eq 1 ]; then
            echo ${a[@]:1}
            continue
        fi
        echo ${a[@]}|cut -d" " -f1,${j-1},${j+1}-
    done <$file
}

```

```
file=$1
n_rows=$(wc -l <$file)
n_cols=$(num_of_cols $file)
flag=NO
for i in $(seq $n_rows); do
  for j in $(seq $n_cols); do
    kiluf_i_j $file $i $j >| tmp
    flag=$(is_sum_two_cols_equal tmp)
    if [ $flag = YES ]; then
      break
    fi
  done
  if [ $flag = YES ]; then
    break
  fi
done
echo $flag
```