

יטודות מערכות פתוחות
פתרון תרגיל מס' 10

שימו לב: כל ההערות שבתחילת תרגילים 9-1 תקפות גם לתרגיל זה.

1. שאלה זו מבוססת על שאלה שהופיעה במבחן מועד א 2016

הגדרה: נגדיר שמטריצה (לא בהכרח ריבועית ולא בהכרח עם מספר שווה של איברים בכל שורה) היא טובה אם האיברים בכל שורה במטריצה יוצרים סדרה חשבונית.

לדוגמה, הקבצים F1 C F2 הם טובים והקבצים A B אינם טובים:

F1	C	F2	A	B
10 20	400	10	15 20 25	1 2 3 1
1		2 4 6 8 10	2 2 2 4	
1 2 3 4				
8 4 0				

כתוב/כתבי תכנית ב- awk בשם P10.1 שמקבלת כפרמטרים רשימת שמות קבצים שמכילים מטריצות של מספרים (לא בהכרח ריבועיות ולא בהכרח עם מספר שווה של איברים בכל שורה), ומדפיסה לפלט שורה אחת עבור כל קובץ שמכילה את שם הקובץ, לאחריו תו רווח ולאחריו YES אם הקובץ טוב או NO אם הקובץ אינו טוב.

על שורות הפלט להופיע לפי הסדר הבא: תחילה הקבצים הטובים (ממוינים לפי סדר לכסיקוגרפי עולה) ולאחר מכן הקבצים שאינם טובים (ממוינים לפי סדר לכסיקוגרפי עולה).

בתכנית P10.1 מותר להשתמש לכל היותר ב- פקודת system אחת.

לדוגמה, לאחר הפעלת התכנית על ידי הפקודה:

P10.1 A B C F1 F2

יתקבל הפלט:

C YES
F1 YES
F2 YES
A NO
B NO

פתרון שאלה 10.1

```
#!/bin/awk -f
function sort1(file) {
    system("sort "file)
}

function check_sidra(sidra,n,A,diff,i) {
    n=split(sidra,A," ")
    if (n<=2) return "YES"
    diff=A[2]-A[1]
    for (i=3;i<=n;i++) {
        if ((A[i]-A[i-1])!=diff) return "NO"
    }
    return "YES"
}

function check(file,result,sidra,i) {
    result="good"
    while (getline<file) {
        sidra=""
        for (i=1;i<=NF;i++) {
            sidra=sidra " "$i
        }
        if (check_sidra(substr(sidra,2))!="YES") {
            result="bad"
            break
        }
    }
    close(file)
    return result
}

BEGIN {
    num_good=0;num_bad=0
    for (i=1;i<ARGC;i++) {
        if (check(ARGV[i])== "good") {
            print ARGV[i] " YES" > "good_files"; num_good++ }
        else {
            print ARGV[i] " NO" > "bad_files"; num_bad++
        }
    }
    if (num_good > 0) sort1("good_files")
    if (num_bad > 0) sort1("bad_files")
}
```

להלן אותו פתרון בפורמט טקסט.

```
#!/bin/awk -f
function sort1(file){
    system("sort "file)
}

function check_sidra(sidra,n,A,diff,i){
    n=split(sidra,A," ")
    if (n<=2) return "YES"
    diff=A[2]-A[1]
    for (i=3;i<=n;i++){
        if ((A[i]-A[i-1])!=diff) return "NO"
    }
    return "YES"
}

function check(file,result,sidra,i){
    result="good"
    while (getline<file){
        sidra=""
        for (i=1;i<=NF;i++){
            sidra=sidra" "$i
        }
        if (check_sidra(substr(sidra,2))!="YES") {
            result="bad"
            break
        }
    }
    close(file)
    return result
}

BEGIN {
    num_good=0;num_bad=0
    for (i=1;i<ARGC;i++) {
        if (check(ARGV[i])=="good") {
            print ARGV[i]" YES" > "good_files"; num_good++ }
        else {
            print ARGV[i]" NO" > "bad_files"; num_bad++
        }
    }
    if (num_good > 0) sort1("good_files")
    if (num_bad > 0) sort1("bad_files")
}
```

2. שאלה זו היא גירסת awk לחלק ראשון של שאלה ב-bash שהופיעה במבחן מועד א 2016.

בספריית בנימינה שומרים נתונים על ספרים בקובץ בשם books שכל שורה שלו היא במבנה הבא:

תאריך החזרה : תאריך השאלה : שם מחבר : שם ספר : מספר עותק

עבור עותק שלא מושאל מופיע - בשדות של תאריך ההשאלה ותאריך ההחזרה. עבור עותק שמושאל ועדין לא מוחזר מופיע - בתאריך ההחזרה.

לדוגמה השורה הבאה:

1234:The Hobbit:J. R. R. Tolkien:1/1/2016:20/1/2016

מצינת שהעותק מספר 1234 של הספר The Hobbit הושאל בתאריך 1/1/2016 והוחזר בתאריך 20/1/2016

השורה הבאה:

1233:The Hobbit:J. R. R. Tolkien:8/1/2016:-

מצינת שהעותק מספר 1233 של הספר The Hobbit הושאל בתאריך 8/1/2016 ועדין לא הוחזר.

השורה הבאה:

1232:The Hobbit:J. R. R. Tolkien:-:-

מצינת שהעותק מספר 1232 של הספר The Hobbit נמצא כרגע בספריה ואינו מושאל.

כתוב/כתבי תכנית ב- awk בשם P10.2 שמקבלת כפרמטר שם ספר

שם ספר P10.2 available-copy

ומחזירה את כל מספרי העותקים של הספר שנמצאים כרגע בספריה (דהינו אינם מושאלים לאף קורא) בשורה אחת, ממוינים בסדר מספרי עולה. במידה וכל העותקים של הספר מושאלים התכנית מחזירה את ההודעה:

No copies available

מותר להשתמש בפקודת system אחת בלבד לפתרון השאלה.

לדוגמה, נניח שהקובץ books מכיל את הנתונים הבאים:

1234:The Hobbit:J. R. R. Tolkien:1/1/2016:20/1/2016
1232:The Hobbit:J. R. R. Tolkien:--:
1234:The Hobbit:J. R. R. Tolkien:1/2/2016:-
1234:The Hobbit:J. R. R. Tolkien:1/10/2015:5/11/2015
1232:The Hobbit:J. R. R. Tolkien:10/10/2014:1/1/2015
2111:The Da Vinci Code:Dan Brown:20/1/2015:21/2/2015
2111:The Da Vinci Code:Dan Brown:20/3/2015:12/4/2015
1233:The Hobbit:J. R. R. Tolkien:8/1/2016:-
1220:The Hobbit:J. R. R. Tolkien:--:
2112:The Da Vinci Code:Dan Brown:20/1/2016:-
2112:The Da Vinci Code:Dan Brown:20/1/2015:20/2/2015
2111:The Da Vinci Code:Dan Brown:--:
3001:The Shack:William P. Young:10/10/2015:-
1236:The Hobbit:J. R. R. Tolkien:--:
4111:The Fault in Our Stars:John Green:--:
3001:The Shack:William P. Young:25/1/2016:-
5002:Carve the Mark:Veronica Roth:12/1/2017:-
5004:Carve the Mark:Veronica Roth:13/1/2017:-

לאחר הפעלת התכנית על ידי הפקודה:

P10.2 "The Hobbit"

מתקבל הפלט:

1220 1232 1236

לאחר הפעלת התכנית על ידי הפקודה:

P10.2 "The Shack"

מתקבל הפלט:

No copies available

```
#!/bin/awk -f
BEGIN {
    split("", COPIES)
    while (getline<"books"){
        split($0,A,":")
        id=A[1];book=A[2];borrow_date=A[4];return_date=A[5]
        if (book != ARGV[1]) continue
        if (borrow_date != "-" && return_date == "-") {
            COPIES[id]="not_available"
        }
        if (COPIES[id] != "not_available") {
            COPIES[id] = "available"
        }
    }
    printf "" > "tmp"
    for (id in COPIES) {
        if ( COPIES[id] == "available") {
            print id > "tmp"
        }
    }
    close("tmp")
    system("sort -n tmp >| tmp1")
    result=""
    while (getline<"tmp1"){
        result=result" "$0
    }

    if (result != "") {
        print substr(result,2)}
    else {
        print "No copies available"
    }
}
```

להלן אותו פתרון בפורמט טקסט.

```
#!/bin/awk -f
BEGIN {
    split("",COPIES)
    while (getline<"books"){
        split($0,A,":")

        id=A[1];book=A[2];borrow_date=A[4];return_date=A[5]
        if (book != ARGV[1]) continue
        if (borrow_date != "-" && return_date == "-") {
            COPIES[id]="not_available"
        }
        if (COPIES[id] != "not_available") {
            COPIES[id] = "available"
        }
    }
    printf "" > "tmp"
    for (id in COPIES) {
        if ( COPIES[id] == "available") {
            print id > "tmp"
        }
    }
    close("tmp")
    system("sort -n tmp >| tmp1")
    result=""
    while (getline<"tmp1"){
        result=result" "$0
    }
    if (result != "") {
        print substr(result,2)}
    else {
        print "No copies available"
    }
}
```

3. שאלה זו היא גירסת awk לחלק שני של שאלה ב- bash שהופיעה במבחן מועד א 2016.

בהמשך לתאור הנתונים של הקובץ books בשאלה 2 כתוב/כתבי תכנית ב- awk בשם P10.3 שמדפיסה את שמות הספרים שהושאלו, כל שם ספר שהושאל מופיע בשורה נפרדת, שמכילה את שם הספר, לאחר מכן תו רווח אחד ולאחר מכן מספר השאלות של הספר. על שורות הפלט להיות ממוינות לפי מספר ההשאלות בסדר מספרי יורד. שורות שמתארות ספרים עם מספר השאלות זהה, ממוינות לפי שמות הספרים בסדר לכסיקוגרפי עולה.

מותר להשתמש בכלל היותר 2 פקודות system לפתרון השאלה.

לדוגמה, עבור הקובץ books שתואר בשאלה 2 לאחר הפעלת התכנית על ידי הפקודה P10.3 יתקבל הפלט הבא:

The Hobbit 5
The Da Vinci Code 4
Carve the Mark 2
The Shack 2


```
#!/bin/awk -f
BEGIN {
    split("",BORROWS)
    while (getline<"books"){
        split($0,A,":")
        book=A[2];borrow_date=A[4]
        if (borrow_date == "-") continue
        BORROWS[book]+=1
    }
    printf "" > "tmp"
    for (book in BORROWS) {
        print BORROWS[book] " "book>"tmp"
    }
    close("tmp")
    system("sort -k 1nr,1 -k 2 tmp >| tmp1")
    while (getline<"tmp1"){
        for (i=2;i<=NF;i++){
            printf $i" "
        }
        printf $1"\n"
    }
}
```

להלן אותו פתרון בפורמט טקסט.

```
#!/bin/awk -f
BEGIN {
    split("",BORROWS)
    while (getline<"books"){
        split($0,A,":")
        book=A[2];borrow_date=A[4]
        if (borrow_date == "-") continue
        BORROWS[book]+=1
    }
    printf "" > "tmp"
    for (book in BORROWS) {
        print BORROWS[book] " "book>"tmp"
    }
    close("tmp")
    system("sort -k 1nr,1 -k 2 tmp >| tmp1")
    while (getline<"tmp1"){
        for (i=2;i<=NF;i++){
            printf $i" "
        }
        printf $1"\n"
    }
}
```