

יטודות מערכות פתוחות
פתרון תרגיל מס' 11

שימו לב: כל ההערות שבתחילת תרגילים 1-10 תקפות גם לתרגיל זה.

1. שאלה זו הופיעה 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 בספרייה ואינו מושאל.

כתוב/כתבי תכנית ב- bash בשם P11.1 (אין להשתמש בפקודות awk ו- sed בשאלה זו) שמקבלת כפרמטרים את אחת מהאפשרויות הבאות ומגיבה בהתאם כפי שמתואר עבור כל אפשרות:

שם ספר P11.1 available-copy

התכנית מחזירה את כל מספרי העותקים של הספר שנמצאים כרגע בספרייה (דהינו אינם מושאלים לאף קורא) בשורה אחת, ממוינים

בסדר מספרי עולה. במידה וכל העותקים של הספר מושאלים התכנית
מחזירה את ההודעה:

No copies available

P11.1 popular-books-report

התכנית מדפיסה 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.

P11.1 return-copy מספר עותק

התכנית מעדכנת את הקובץ books שהעותק של הספר חזר בתאריך שבו
מופעלת הפקודה.

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:10/10/2015:10/2/2016
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:

P11.1 available-copy "The Hobbit"

מתקבל הפלט:

1220 1232 1236

לאחר הפעלת התכנית על ידי הפקודה:

P11.1 available-copy "The Shack"

מתקבל הפלט:

No copies available

לאחר הפעלת התכנית על ידי הפקודה:

P11.1 popular-books-report

מתקבל הפלט:

The Hobbit 5

The Da Vinci Code 4

Carve the Mark 2

The Shack 2

לאחר הפעלת התכנית על ידי הפקודה:

P11.1 return-copy 5002

השורה הלפני אחרונה שבקובץ books הנ"ל מוחלפת בשורה הבאה
(שמכילה את התאריך של היום במקום של תאריך ההחזרה):

5002:Carve the Mark:Veronica Roth:12/1/2017:25/1/2017

התאריך של היום נקבע על ידי הפעלת הפקודה date ומתוך הפקודה
שליפת היום החודש והשנה. מאחר והחודש כתוב בצורה מילולית (של 3
אותיות) יש להחליף אותו למספר המתאים.

לדוגמה בזמן כתיבת השאלה הפלט של הפקודה date היה:

Wed Jan 25 12:17:58 IST 2017

פתרון שאלה 11.1

```
function available-copy {
  >|non_avail; >|avail
  local book1="$1" book id borrow_date return_date line
  while read line; do
    book=$(echo $line | cut -d":" -f2)
    id=$(echo $line | cut -d":" -f1)
    borrow_date=$(echo $line | cut -d":" -f4)
    return_date=$(echo $line | cut -d":" -f5)
    if [ "$book" != "$book1" ];then
      continue
    fi
    if [ "$borrow_date" != "-" -a "$return_date" == "-" ]
    then
      echo $id >> non_avail; continue
    fi
    echo $id >> avail
  done<books
  >|available-copies
  while read id; do
    if [ $(egrep -c "^$id$" non_avail) -eq 0 ];then
      echo $id >> available-copies
    fi
  done<avail
  if [ $(wc -l < available-copies) -eq 0 ]; then
    echo "No copies available"
  else
    echo $(sort -nu available-copies)
  fi
}

function calc_book {
  local book1="$1" book n=0 borrow_date line
  while read line; do
    book=$(echo "$line" | cut -d":" -f2)
    borrow_date=$(echo "$line" | cut -d":" -f4)
    if [ "$book" != "$book1" ];then
      continue
    fi
    if [ "$borrow_date" != "-" ];then
      ((n+=1))
    fi
  done<books
  echo $n
}
```

```

function popular-books-report {
    local book n; >|tmp1
    cat books|cut -d":" -f2 |sort -u >|tmp
    while read book; do
        n=$(calc_book "$book")
        if [ $n -ne 0 ]; then
            echo "$n $book $n" >>tmp1
        fi
    done<tmp
    sort -k 1nr,1 -k 2 tmp1 | cut -d" " -f2-
}

function get_date {
    local year month day
    year=$(date|cut -d" " -f6)
    month=$(date|cut -d" " -f2)
    day=$(date|cut -d" " -f3)
    echo Jan 1 >|months
    echo Feb 2 >>months
    echo Mar 3 >>months
    echo Apr 4 >>months
    echo May 5 >>months
    echo Jun 6 >>months
    echo Jul 7 >>months
    echo Aug 8 >>months
    echo Sep 9 >>months
    echo Oct 10 >>months
    echo Nov 11 >>months
    echo Dec 12 >>months
    month=$(egrep "^$month " months |cut -d" " -f2)
    echo "$day/$month/$year"
}

function return-copy {
    local id id1 borrow_date return_date line x
    id1=$1; >|tmp1
    while read line; do
        id=$(echo $line|cut -d":" -f1)
        borrow_date=$(echo $line|cut -d":" -f4)
        return_date=$(echo $line|cut -d":" -f5)
        if [ $id -ne $id1 ]; then
            echo "$line" >> tmp1
            continue
        fi
        if [ "$return_date" != "-" -o $borrow_date == "-" ]
        then
            echo "$line" >> tmp1
            continue
        fi
        date1=$(get_date)
        x=$(echo "$line" |cut -d":" -f1-4)
        echo "${x}:${date1}" >> tmp1
    done<books
    cat tmp1 >|books
}

```

```

if [ "$1" == "available-copy" ]; then
    available-copy "$2"
elif [ "$1" == "popular-books-report" ]; then
    popular-books-report
elif [ "$1" == "return-copy" ]; then
    return-copy "$2"
fi

```

להלן אותו פתרון בפורמט טקסט.

```

function available-copy {
    >|non_avail; >|avail
    local book1="$1" book id borrow_date return_date line
    while read line; do
        book=$(echo $line | cut -d":" -f2)
        id=$(echo $line | cut -d":" -f1)
        borrow_date=$(echo $line | cut -d":" -f4)
        return_date=$(echo $line | cut -d":" -f5)
        if [ "$book" != "$book1" ];then
            continue
        fi
        if [ "$borrow_date" != "-" -a "$return_date" == "-" ]
            then
                echo $id >> non_avail; continue
            fi
            echo $id >> avail
    done<books
    >|available-copies
    while read id; do
        if [ $(egrep -c "^$id$" non_avail) -eq 0 ];then
            echo $id >> available-copies
        fi
    done<avail
    if [ $(wc -l < available-copies) -eq 0 ]; then
        echo "No copies available"
    else
        echo $(sort -nu available-copies)
    fi
}

```

```

function calc_book {
    local book1="$1" book n=0 borrow_date line
    while read line; do
        book=$(echo "$line" | cut -d":" -f2)
        borrow_date=$(echo "$line" | cut -d":" -f4)
        if [ "$book" != "$book1" ];then
            continue
        fi
        if [ "$borrow_date" != "-" ];then
            ((n+=1))
        fi
    done<books
    echo $n
}

function popular-books-report {
    local book n; >|tmp1
    cat books|cut -d":" -f2 |sort -u >|tmp
    while read book; do
        n=$(calc_book "$book")
        if [ $n -ne 0 ]; then
            echo "$n $book $n" >>tmp1
        fi
    done<tmp
    sort -k 1nr,1 -k 2 tmp1 | cut -d" " -f2-
}

function get_date {
    local year month day
    year=$(date|cut -d" " -f6)
    month=$(date|cut -d" " -f2)
    day=$(date|cut -d" " -f3)
    echo Jan 1 >|months
    echo Feb 2 >>months
    echo Mar 3 >>months
    echo Apr 4 >>months
    echo May 5 >>months
    echo Jun 6 >>months
    echo Jul 7 >>months
    echo Aug 8 >>months
    echo Sep 9 >>months
    echo Oct 10 >>months
    echo Nov 11 >>months
    echo Dec 12 >>months
    month=$(egrep "^$month " months |cut -d" " -f2)
    echo "$day/$month/$year"
}

```

```

function return-copy {
    local id id1 borrow_date return_date line x
    id1=$1; >|tmp1
    while read line; do
        id=$(echo $line|cut -d":" -f1)
        borrow_date=$(echo $line|cut -d":" -f4)
        return_date=$(echo $line|cut -d":" -f5)
        if [ $id -ne $id1 ]; then
            echo "$line" >> tmp1
            continue
        fi
        if [ "$return_date" != "-" -o $borrow_date == "-" ]
        then
            echo "$line" >> tmp1
            continue
        fi
        date1=$(get_date)
        x=$(echo "$line" |cut -d":" -f1-4)
        echo "${x}:${date1}" >> tmp1
    done<books
    cat tmp1 >|books
}
if [ "$1" == "available-copy" ]; then
    available-copy "$2"
elif [ "$1" == "popular-books-report" ]; then
    popular-books-report
elif [ "$1" == "return-copy" ]; then
    return-copy "$2"
fi

```

.2

ברשת הסניפים של רמי לוי שומרים נתונים על רכישות מוצרים כפי שמתואר להלן.

עבור כל סניף ברשת שומרים בקובץ (ששמו כשם הסניף) את הנתונים הבאים:

מספר קופה, מספר פריט, כמות פריטים, מחיר ליחידה, תאריך מכירה. לדוגמה נניח שיש שורה בקובץ ששמו `zichron` שמכילה:

```
3 1234 2 30 20/2/2014
```

אז המשמעות היא שבקופה מספר 3 בתאריך 20/2/2014 התבצעה מכירה של 2 יחידות של פריט שמספרו 1234 כאשר מחיר כל פריט היה 30 ש"ח.

כתוב תוכנית `Script` ב-`Bash` בשם `P11.2` (אין להשתמש בפקודות `awk` ו-`sed` בשאלה זו) שמקבלת כפרמטרים שני תאריכים ולאחריהם רשימת שמות קבצים שמתארים את הנתונים של סניפים ברשת הסניפים של רמי לוי.

הפורמט של שורות הקבצים שמתארים את הסניפים הוא כפי שתואר לעיל.

התכנית `P11.2` מדפיסה לפלט שורה אחת עבור כל פריט שנמכר באיזשהו סניף בין התאריך הראשון לתאריך השני שהועברו כפרמטרים לתכנית (כולל התאריך הראשון והתאריך השני). השורה מכילה את מספר הפריט ולאחר מכן נתונים על המכירות של הפריט בתאריכים הנדרשים. הפורמט של הפלט הוא כפי שמתואר בדוגמה שבהמשך.

על שורות הפלט להיות ממוינות לפי מספרי פריט בסדר עולה, ובתוך כל שורה על נתוני הסניפים להיות ממוינים לפי סך כל המכירות של הפריט בתאריכים הנדרשים בסדר יורד.

סניף שבו לא נמכר הפריט בתאריכים הנדרשים, לא יופיע בשורה של הפריט.

פריט שלא נמכר בתאריכים הנדרשים באף סניף לא יופיע בפלט.

לדוגמה,

נניח שנתונים 3 הקבצים הבאים:

נניח שתוכן הקובץ haifa הוא:

1	4567	1	40	20/12/2016
4	2222	2	20	21/10/2016
4	4567	3	30	20/2/2016
2	2222	1	20	12/1/2017
2	1111	2	60	21/1/2017
2	3333	2	40	2/10/2016

נניח שתוכן הקובץ tel-aviv הוא:

1	4567	2	30	20/12/2016
4	1111	10	20	1/12/2016
1	4567	3	30	20/5/2016
1	1111	2	30	20/5/2016

נניח שתוכן הקובץ jerusalem הוא:

1	2222	4	30	20/1/2017
4	1111	3	20	23/1/2017
2	4567	3	40	21/1/2017

לאחר הפעלת התכנית P11.2 על ידי הפקודה:

P11.2 20/10/2016 21/1/2017 haifa tel-aviv jerusalem

מתקבל הפלט:

1111 tel-aviv 200 haifa 120
2222 jerusalem 120 haifa 60
4567 jerusalem 120 tel-aviv 60 haifa 40

פתרון שאלה 11.2

```
function check_date1_le_date2 {
    local day1 day2 month1 month2 year1 year2
    day1=$(echo $1 | cut -d"/" -f1)
    day2=$(echo $2 | cut -d"/" -f1)
    month1=$(echo $1 | cut -d"/" -f2)
    month2=$(echo $2 | cut -d"/" -f2)
    year1=$(echo $1 | cut -d"/" -f3)
    year2=$(echo $2 | cut -d"/" -f3)
    if [ $year1 -gt $year2 ]; then
        echo "NO"
    elif [ $year1 -lt $year2 ]; then
        echo "YES"
    elif [ $month1 -gt $month2 ]; then
        echo "NO"
    elif [ $month1 -lt $month2 ]; then
        echo "YES"
    elif [ $day1 -gt $day2 ]; then
        echo "NO"
    else
        echo "YES"
    fi
}

function create_data_for_correct_dates {
    local date1=$1 date2=$2 sniff=$3 line y1 y2
    while read line; do
        date=$(echo $line | cut -d" " -f5)
        if [ $(check_date1_le_date2 $date1 $date) == "NO" ]
        then
            continue
        fi
        if [ $(check_date1_le_date2 $date $date2) == "NO" ]
        then
            continue
        fi
        parit=$(echo $line | cut -d" " -f2)
        y1=$(echo $line | cut -d" " -f3)
        y2=$(echo $line | cut -d" " -f4)
        echo "$sniff $parit ${y1*y2}" >> all_data
    done<$sniff
}
```

```

function calc_parit_snif {
    local parit=$1; snif=$2
    egrep "^${snif} ${parit} " all_data|cut -d" " -f3>|tmp1
    if [ $(wc -l < tmp1) -gt 0 ];then
        echo ${$(echo $(cat tmp1) | tr " " "+")}
    else
        echo 0
    fi
}

date1=$1;date2=$2;
shift;shift
>|all_data
for snif; do
    create_data_for_correct_dates $date1 $date2 $snif
done
cut -d" " -f1 all_data | sort -u >| snifim
cut -d" " -f2 all_data | sort -nu >| pritim
for parit in $(echo $(cat pritim)); do
    >|tmp
    for snif in $(echo $(cat snifim)); do
        total=$(calc_parit_snif $parit $snif)
        if [ $total -ne 0 ]; then
            echo $snif $total >> tmp
        fi
    done
    echo $parit $(sort -k 2nr,2 tmp | tr "\n" " ")
done

```

להלן אותו פתרון בפורמט טקסט.

```
function check_date1_le_date2 {
    local day1 day2 month1 month2 year1 year2
    day1=$(echo $1 | cut -d"/" -f1)
    day2=$(echo $2 | cut -d"/" -f1)
    month1=$(echo $1 | cut -d"/" -f2)
    month2=$(echo $2 | cut -d"/" -f2)
    year1=$(echo $1 | cut -d"/" -f3)
    year2=$(echo $2 | cut -d"/" -f3)
    if [ $year1 -gt $year2 ]; then
        echo "NO"
    elif [ $year1 -lt $year2 ]; then
        echo "YES"
    elif [ $month1 -gt $month2 ]; then
        echo "NO"
    elif [ $month1 -lt $month2 ]; then
        echo "YES"
    elif [ $day1 -gt $day2 ]; then
        echo "NO"
    else
        echo "YES"
    fi
}

function create_data_for_correct_dates {
    local date1=$1 date2=$2 snif=$3 line y1 y2
    while read line; do
        date=$(echo $line | cut -d" " -f5)
        if [ $(check_date1_le_date2 $date1 $date) == "NO" ]
        then
            continue
        fi
        if [ $(check_date1_le_date2 $date $date2) == "NO" ]
        then
            continue
        fi
        parit=$(echo $line | cut -d" " -f2)
        y1=$(echo $line | cut -d" " -f3)
        y2=$(echo $line | cut -d" " -f4)
        echo "$snif $parit ${y1*y2}" >> all_data
    done<$snif
}
```

```

function calc_parit_snif {
    local parit=$1; snif=$2
    egrep "^${snif} ${parit} " all_data|cut -d" " -f3>|tmp1
    if [ $(wc -l < tmp1) -gt 0 ];then
        echo [$$(echo $(cat tmp1) | tr " " "+")]
    else
        echo 0
    fi
}

date1=$1;date2=$2;
shift;shift
>|all_data
for snif; do
    create_data_for_correct_dates $date1 $date2 $snif
done
cut -d" " -f1 all_data | sort -u >| snifim
cut -d" " -f2 all_data | sort -nu >| pritim
for parit in $(echo $(cat pritim)); do
    >|tmp
    for snif in $(echo $(cat snifim)); do
        total=$(calc_parit_snif $parit $snif)
        if [ $total -ne 0 ]; then
            echo $snif $total >> tmp
        fi
    done
    echo $parit $(sort -k 2nr,2 tmp | tr "\n" " ")
done

```