

20.2.2017

ביה"ס למדעי המחשב ומתמטיקה
המכללה האקדמית נתניה

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

- משך המבחן: שלוש וחצי שעות.
- יש לענות על כל השאלות.
- מותר השימוש בחומר עזר כלשהו, פרט למחשבים, (מחשבונים מותר).
- יש להקפיד על כתיבה ברורה ומסודרת של התשובות.

1.15 (נקודות)

חלק א (5 נקודות)

נניח שנתון קובץ בשם P1.1 שתכנו הוא:

```
echo -n "Enter number: "  
read num  
echo -n "Enter string: "  
read str  
res=""  
n=$(echo -n $str | wc -c)  
for i in $(seq $num); do  
    c1=$(echo $str | cut -c$i)  
    c2=$(echo $str | cut -c${n+1-i})  
    res=${c2}${res}${c1}  
done  
echo "The result is: $res"
```

ונניח שנתון קובץ בשם P1.2 שתכנו הוא:

```
#!/usr/bin/expect  
set x [lindex $argv 0]  
set y [lindex $argv 1]  
log_user 0  
spawn P1.1  
log_user 1  
expect "number" {send "$x\n"}  
expect "string" {send "$y\n"}  
expect "The" {exit}
```

מה יתקבל בפלט (דהינו יוצג על המסך) לאחר הפעלת הפקודה:

P1.2 3 abcdef

חלק ב (10 נקודות)

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

```
echo "output 1:"
x="a d f"
echo [$$(echo $x|tr a-f 1-6|tr " " "+")]

echo "output 2:"
echo "ab@% cd&!xz" | tr -cd "a-z\n"

echo "output 3:"
echo "abchiaza" >|F1
echo "aa hiz abbza" >> F1
echo "aabzbb lhi " >> F1
echo " a2hz xi" >>F1
egrep "(^a|[ ]+a)([^ a]+)z([ ]|a$)" F1

echo "output 4:"
echo abc |egrep -o .| awk '{getline;print $1}'

echo "output 5:"
echo "abaccb"|awk '{gsub("([ab]+)","&z&",$0);print $0}'
```

2. (10 נקודות)

נגדיר שמילה היא רצף של תווים ללא תווי רווח וסוף שורה.

כתוב תכנית Script ב- sed בשם p2 שמקבלת כפרמטר שם קובץ ומדפיסה את הפלט שמתקבל באופן הבא:
אם בשורה בקובץ יש פחות משתי מילים אז השורה לא מופיעה בפלט.

אם בשורה בקובץ במילה השנייה בשורה יש פחות מ- 3 תווים אז השורה לא מופיעה בפלט.

אם בשורה בקובץ במילה השנייה יש לפחות 3 תווים אז מופיעה בפלט השורה שמתקבלת לאחר שכפול 3 התווים הראשונים במילה השנייה בשורה, כפי שמתואר בדוגמה שבהמשך.

בסוף הפלט מופיע מספר שמציין את מספר השורות בקובץ שלא הופיעו בפלט.

על המבנה של התכנית p2 להיות כדלהלן:

התכנית מכילה שורה אחת או יותר של פקודות sed לדוגמה:

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

אין להשתמש בפקודות שאינן של sed ושאיןן 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
dea 123 123 zy45
12 12
abc cd abc cd
xyz%$@! x@yz%$@! kk
xyzxyz
abc htzw
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P2 F2
```

יתקבל הפלט:

```
dea 112233 123 zy45
xyz%$@! xx@@yyz%$@! kk
abc hhttzzw
4
```

3. (25 נקודות)

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 הושאל אף פעם מופיע - (מינוס) בשדות של תאריך ההשאלה ותאריך ההחזרה.

עבור עותק שהושאל ועדיין לא הוחזר מופיע - (מינוס) בתאריך ההחזרה.

עותק מסוים נחשב פנוי אם אין אף שורה בקובץ שמצינת שהעותק הושאל ועדיין לא הוחזר.

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 הנ"ל לא בהכרח מצינת שהעותק 1234 פנוי כי יתכן ויש שורה נוספת בקובץ שמצינת שהוא הושאל (לאחר מכן) ועדיין לא הוחזר.

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 ועדיין לא הוחזר.

השורה הבאה:

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

מצינת שהעותק מספר 1232 של הספר The Hobbit נמצא כרגע בספריה ולא הושאל אף פעם.

כתוב/כתבי תכנית ב- awk בשם p3 שמקבלת כפרמטר שם סופר ומחזירה את כל שמות כל הספרים של הסופר הזה שיש מהם לפחות עותק אחד פנוי בספריה. על כל שם ספר להופיע פעם אחת בדיוק בפלט ובשורה נפרדת ועל שמות הספרים להיות ממוינים לפי סדר לכסיגוקרפי עולה.

במידה ואין כלל ספרים שעבורם יש עותקים פנויים בספריה עבור סופר זה תופיע ההודעה:

No available books for this author

מותר להשתמש בפקודת system אחת בלבד לפתרון השאלה.

לדוגמה, נניח שהקובץ books מכיל את הנתונים הבאים:

1234:Crime and Punishment:F. Dostoevsky:1/1/2016:20/1/2016
1232:The Hobbit:J. R. R. Tolkien:--
1234:Crime and Punishment:F. Dostoevsky: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:Crime and Punishment:F. Dostoevsky:20/1/2016:-
2111:Crime and Punishment:F. Dostoevsky:20/3/2015:12/4/2015
1233:The Hobbit:J. R. R. Tolkien:8/1/2016:-
1220:The Gambler:F. Dostoevsky:--
2112:The Gambler:F. Dostoevsky:20/1/2016:-
2112:The Gambler:F. Dostoevsky:20/1/2015:20/2/2015
3007:The Shack:William P. Young:10/10/2015:-
1236:The Hobbit:J. R. R. Tolkien:--
4111:The Idiot:F. Dostoevsky:20/1/2017:20/2/2017
3001:The Shack:William P. Young:25/1/2016:-
5002:The Brothers Karamazov:F. Dostoevsky:12/1/2017:-
5004:The House of the Dead:F. Dostoevsky:13/1/2017:12/2/2017

לאחר הפעלת התכנית על ידי הפקודה:

P3 "F. Dostoevsky"

מתקבל הפלט:

The Gambler
The House of the Dead
The Idiot

שימו לב שהספרים Crime and Punishment
ו- The Brothers Karamazov לא מופיעים בפלט כי אין עותקים
פנויים שלהם בספרייה.

לאחר הפעלת התכנית על ידי הפקודה:

P3 "William P. Young"

מתקבל הפלט:

No available books for this author

4. (25 נקודות)

נגדיר שמילה נמצאת בקובץ בתחום המוגדר על ידי 4 מספרים $j_2 j_1 i_2 i_1$ אם המילה נמצאת בקובץ בין שורה i_1 לבין שורה i_2 (כולל שורות $i_1 - 1$) ובין עמודות $j_1 - 1$ ו- j_2 (כולל עמודות $j_1 - 1$ ו- j_2).

לדוגמה עבור הקובץ הבא:

```
ab    cd    ef  gh
12  xy    zw
67 89 zzz ddd
```

המילה 89 נמצאת בתחום המוגדר על ידי 4 המספרים 2 3 1 2. המילה zzz אינה נמצאת בתחום המוגדר על ידי 4 המספרים 2 3 1 2.

כתוב/כתבי תכנית ב- `bash` בשם `P4` (אין להשתמש בפקודות `awk` ו-`sed` בשאלה זו) שמקבלת כפרמטרים 4 מספרים $j_2 j_1 i_2 i_1$ ולאחריהם רשימת שמות קבצים ומדפיסה לפלט את כל המילים שנמצאות בתחום שמוגדר על ידי 4 המספרים $j_2 j_1 i_2 i_1$ בכל אחד מהקבצים שברשימת הקבצים. כל מילה תודפס בשורה נפרדת שמכילה את המילה תו רווח אחד ולאחר מכן את מספר ההופעות הכולל של המילה בקבצים שברשימת הקבצים (לא בהכרח בתחום המוגדר על ידי המספרים).

על שורות הפלט להיות ממוינות לפי מספר ההופעות הכולל של המילים בסדר מספרי יורד. (עבור מילים עם אותו מספר הופעות אין חשיבות לסדר בין מילים אלו).

ראה/י דוגמה בעמוד הבא.

לדוגמה, נניח שנתונים הקבצים הבאים:

קובץ H שתוכנו הוא:

```
ab gh cd gh
12 xy ef ef
66 89 zzz ddd
3 44 55 66 ddd
1 66 3 4 66 ef
7 8 9 88 99 gh
```

קובץ G שתוכנו הוא:

```
ab cd ef gh 66
12 xy zw
67 uu zzz ef
3 99 55 66 88 ddd 99
```

קובץ I שתוכנו הוא:

```
ab cd ef gh
12 xy zw 88
67 89 zzz ddd 88
3 44 55 66 77
a b c d ef
```

לאחר הפעלת התכנית על ידי הפקודה:

```
P4 3 6 4 6 H G I
```

מתקבל הפלט:

```
66 7
ef 6
ddd 4
88 4
```


5.25 נקודות

נגדיר שקובץ הוא מטריצה ריבועית אם בכל שורה בקובץ יש אותו מספר מילים ומספר השורות בקובץ שווה למספר המילים בשורה.

לדוגמה, הקבצים H1 H2 הם מטריצות ריבועיות והקבצים G1 G2 אינם מטריצות ריבועיות:

H1	H2	G1	G2
10 20 ab	10	10 20	ab cd
-30 cd 10			jj 12 34
a b c			

כתוב/כתבי תכנית ב-Bash (דהינו קובץ Script) בשם P5 שמקבלת כפרמטרים רשימת תיקיות ומדפיסה לפלט את שמות כל הקבצים שמכילים מטריצות ריבועיות ונמצאים באחת מהתיקיות (בעומק כלשהו). כל שם קובץ יופיע בשורה נפרדת ויתאר את המסלול אל הקובץ החל מהתיקה הנוכחית בה מריצים את התכנית P5. על סדר הקבצים בפלט להיות לפי השמות הסופיים של הקבצים בסדר לכסיקוגרפי עולה. (אין חשיבות לסדר בין קבצים עם שם סופי זהה).

ראה/י דוגמה בעמוד הבא.

לדוגמה, להלן מבנה תיקיות d1 d2 שנמצאות בתיקיה הנוכחית בה מריצים את התכנית P5. המבנה מתואר כפי שמתקבל על ידי הפעלת הפקודה tree על תיקיות אלה, כאשר מימין לכל קובץ מופיע y או n בהתאם לכך אם הקובץ מכיל מטריצה ריבועית או לא. (התוספת הזו היא רק לצורך ההסבר של הדוגמה ואינה מופיעה כפלט של הפקודה tree).

```
d1
|-- A  n
|-- d2
|   |-- A  y
|   |-- CC y
|   |-- F1 n
|   `-- F2 y
|-- CC  n
|-- F1  y
`-- d3
    `-- F1 n
```

```
d2
|-- A  y
|-- d4
|   |-- B  y
|   |-- d5
|   |   |-- A  y
|   |   `-- CC y
|   |-- F1  n
|   `-- F2  n
`-- CC  n
```

לאחר הפעלת התכנית ע"י הפקודה:

```
P5 d1 d2
```

יתקבל הפלט:

```
d1/d2/A
d2/A
d2/d4/d5/A
d2/d4/B
d1/d2/CC
d2/d4/d5/CC
d1/F1
d1/d2/F2
```

! בהצלחה