

19.2.2013

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

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

1. (15 נקודות)
חלק א' (5 נקודות)

נתונים קבצים בשם P1 ו P2 - כאשר תוכן הקובץ P1 הוא:

```
#!/bin/bash
echo 12345 >| F1
echo 5221 >> F1
echo 33 >> F1
cat F1
```

ותוכן הקובץ P2 הוא:

```
#!/bin/bash
while read x
do
n=$(echo -n "$x" | wc -c)
s=0
for i in $(seq $n)
do
y=$(echo "$x" | cut -c${i})
s=${s+$y}
done
echo "$x" $s
done
```

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

P1 | P2

חלק ב (10 נקודות)

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

```
echo "output 1:"
echo -e "abcd\n1234\nxyz" >| F1
echo -e "1\n2\n3\n4" >| F2
echo -e "6\n77\n888\n999" >| F3
cat - F1 F2 < F3 >|F4
cat F1 F4 | wc -l
```

```
echo "output 2:"
echo -n "abc@%^def*ABC1 23" | tr a-z0-9 "\n" | wc -l
```

```
echo "output 3:"
echo "abc shalom" >|F1
echo " abc shalom4" >> F1
echo "abc shalom 123" >> F1
echo "abc shalom123 shalom6" >> F1
echo "abc shalom2 456 shalom" >> F1
egrep "^[ ]*[^ ]+[ ]+shalom($|[123])" F1
```

```
echo "output 4:"
```

```
echo abcdeababcde >| F1
echo 12341112341123411 >> F1
echo aaabbaaaabba >> F1
egrep "^(...+)\2\1$" F1
```

```
echo "output 5:"
```

```
echo "aa xyz abc x" >|F1
echo "cc aa xyz abc" >> F1
echo "abc xyz aa x aa abc abc xyz xyz xyz" >> F1
cat F1 | tr -s " " "\n" | sort | uniq -c | sort -n
```

2. (10 נקודות)

כתוב תוכנית Script - ב sed בשם P2 שמקבלת כפרמטרים שם קובץ (בהמשך נקרא לו קובץ 1) ומחרוזת (בהמשך נקרא לה מחרוזת 1). התוכנית מדפיסה לפלט את השורות בקובץ שמכילות את מחרוזתו (באיזשהו מקום בשורה) כאשר בשורות אלה מתבצע שיכפול של המילה השנייה, כפי שמודגם בדוגמה שלהלן. (ניתן להניח שבכל שורה יש לפחות שתי מילים).

על המבנה של התוכנית P2 להיות כדלהלן:

שורה ראשונה כמו בכל קובץ script ב-bash מכילה:

```
#!/bin/bash
```

לאחר מכן שורה אחת או יותר של פקודות sed לדוגמה

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

(אין להשתמש בפקודות שאינן של sed ושאין 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:

```
abc 1 def2 3 ab4
123 zy
abc1d rq rq
abcabc 12
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P2 F2 abc
```

יתקבל הפלט:

```
abc 1 1 def2 3 ab4
abc1d rq rq rq
abcabc 12 12
```

3. (20 נקודות)

חלק א (15 נקודות)

כתוב/כתבי תוכנית ב- awk בשם P3.1 שמקבלת פרמטר שם קובץ ומדפיסה לפלט מספר שמציין את סכום הספרות הגדול מבין כל המספרים שמופיעים בקובץ.

חלק ב (5 נקודות)

כתוב/כתבי תוכנית ב- awk בשם P3.2 שמקבלת כפרמטרים רשימת שמות של קבצים ומדפיסה לפלט שורה אחת עבור כל קובץ שברשימת הקבצים שמכילה את שם הקובץ, לאחריו תו רווח ולאחריו מספר שמציין את סכום הספרות הגדול ביותר מבין כל המספרים שמופיעים בקובץ. על סדר הקבצים בפלט להיות לפי סדר הופעתם ברשימת הפרמטרים.

לדוגמה, נניח שתוכן הקובץ F1 הוא:

```
1234 99999a 77b22
abcabc 888 99
66666
```

ונניח שתוכן הקובץ A הוא:

```
12yy34 99999a 17522
abcabc 88a8 99
66c666 111
2222222
```

לאחר הפעלת התוכנית P3.1 ע"י הפקודה:

```
awk -f P3.1 F1
```

יתקבל הפלט:

```
30
```

לאחר הפעלת התוכנית P3.2 ע"י הפקודה:

```
awk -f P3.2 F1 A
```

יתקבל הפלט:

```
F1 30
```

```
A 17
```

4. (30 נקודות)

כתוב/כתבי תוכנית ב-bash בשם P4 שמקבלת כפרמטרים מספר (בהמשך נקרא לו n) ולאחריו רשימת שמות של קבצים שמכילים מספרים.

התוכנית מדפיסה לפלט:

בשורה הראשונה :

המספר הגדול ביותר בקובץ הראשון ברשימה, לאחריו תו רווח אחד, לאחריו המספר הגדול ביותר בקובץ השני ברשימה, לאחריו תו רווח אחד, וכן הלאה... עד המספר הגדול ביותר בקובץ האחרון ברשימה.

בשורה השנייה :

המספר השני בגודלו בקובץ הראשון ברשימה, לאחריו תו רווח אחד, לאחריו המספר השני בגודלו בקובץ השני ברשימה, לאחריו תו רווח אחד, וכן הלאה... עד המספר השני בגודלו בקובץ האחרון ברשימה.

וכן הלאה... עד

שורה מספר n שבה יופיע:

המספר ה-n בגודלו בקובץ הראשון ברשימה, לאחריו תו רווח אחד, לאחריו המספר ה-n בגודלו בקובץ השני ברשימה, לאחריו תו רווח אחד, וכן הלאה... עד המספר ה-n בגודלו בקובץ האחרון ברשימה.

הנחות: ניתן להניח שהקבצים מכילים מספרים בלבד (לא בהכרח מספר אחד בכל שורה), ושכל כל המספרים בכל קובץ שונים זה מזה. בנוסף ניתן להניח שבכל קובץ ישנם לפחות n מספרים.

לדוגמה, להלן שמות קבצים ותוכנם.

A	B	C	K	E
10 20	2 3 4	20 200	100	10 20 30 40
30 40	50	60 70	200	
	40 25	3	150	
		400	4	

לאחר הפעלת התוכנית ע"י הפקודה:

P4 3 A B C K E

יתקבל הפלט:

```
40 50 400 200 40
30 40 200 150 30
20 25 70 100 20
```

לאחר הפעלת התוכנית ע"י הפקודה:

P4 4 E C A

יתקבל הפלט:

```
40 400 40
30 200 30
20 70 20
10 60 10
```

5. (25 נקודות)

כתוב/כתבי תוכנית ב-Bash (דהינו קובץ Script) בשם P5 שמקבלת כפרמטרים רשימת תיקיות. התוכנית מדפיסה לפלט את שמות כל הקבצים שמופיעים לפחות פעם אחת (בעומק כלשהו) בכל אחת מהתיקיות שברשימה. כל שם קובץ יופיע בפלט בשורה נפרדת. אין חשיבות לסדר בין שמות הקבצים בפלט.

לדוגמה, להלן מבנה תיקיות d1 d2 d3 כפי שמתקבל על ידי הפעלת הפקודה tree על תיקיות אלו. בתאור המבנה של התיקיות יש להניח ששם שאין מתחתיו קבצים מתאר קובץ (ולא תיקיה).

```
d1
|-- A
|-- AA
| |-- F1
| `-- F2
|-- CC
|-- DDD
`-- EE
    `-- F1
```

```
d2
|-- A1
| |-- F1
| `-- F2
|-- CC
`-- XXX
```

```
d3
|-- B
| |-- F1
| `-- F2
|-- CC
|-- F1
`-- XXX
```

לאחר הפעלת התוכנית ע"י הפקודה:

```
P5 d1 d2 d3
```

יתקבל הפלט:

```
F1
F2
CC
```

בהצלחה !