

יסודות מערכות פתוחות

פתרון תרגיל מס' 3

שימו לב: כל ההערות שבתחילת תרגילים 1-2 תקפות גם לתרגיל זה.

הערות:

1. שגיאה נפוצה בתרגיל זה היא שימוש בהפניית פלט מהסוג > במקום הפניית פלט מהסוג |>. במילים אחרות, יש לדאוג שבתכנית שלכם לא תהינה הפניות פלט מהסוג > ואם השתמשתם בהפניה כזו תחליפו אותה בהפניית פלט מהסוג |>

2. בכל השאלות מותר להשתמש בקבצי ביניים שהתכניות שלכם יוצרות. אם התכנית שלכם משתמשת בקובץ ביניים, למשל F1, יש לוודא בתחילת התכנית שהקובץ F1 הנו ריק. לצורך כך אפשר להשתמש בפקודה הבאה:

```
echo -n "" >| F1
הפקודה הנ"ל גורמת לכך שתוכן הקובץ F1 יהיה ריק ולא יכיל אף תו, (גם לא תו \n).
```

1. כתוב תכנית Script ב-Bash בשם P3.1 שקולטת מהמשתמש שם קובץ שמכיל תכנית (בהמשך נקרא לה program) ולאחר מכן קולטת מהמשתמש מספר (בהמשך נקרא לו n). התכנית P3.1 מדפיסה למסך n עותקים של הפלט הרגיל של התכנית program ולאחר מכן n עותקים של פלט השגיאות של התכנית program.

יש להניח שהתכנית program מופעלת בתיקיה הנוכחית בה מופעלת התכנית P3.1. ניתן להניח שלתכנית program יש הרשאת הרצה.

אין להדפיס הודעות בזמן קליטת שם התכנית והמספר מהמשתמש. (הדפסת הודעות כאלו תחשב כפלט של התכנית P3.1 ותגרום לתכנית הבדיקה האוטומטית להכשיל את התכנית שלכם).

לדוגמה, נניח שתוכן התכנית AA שנמצאת בתיקיה הנוכחית הוא:

```
seq -s" " 8
seq 8 -s" "
echo abc
seq -s"ab" 10 2 16
echo def
x=cat BB
```

נניח שהתכנית P3.1 קולטת מהמשתמש שם תכנית AA ולאחר מכן קולטת מהמשתמש את המספר 2. אזי הפלט של התכנית P3.1 יהיה:

```
1 2 3 4 5 6 7 8
abc
10ab12ab14ab16
def
1 2 3 4 5 6 7 8
abc
10ab12ab14ab16
def
seq: invalid floating point argument: -s
Try `seq --help' for more information.
./AA: line 6: BB: command not found
seq: invalid floating point argument: -s
Try `seq --help' for more information.
./AA: line 6: BB: command not found
```

פתרון שאלה 1

```
read program
read n
rm -f Error
for i in $(seq $n)
do
    $program 2>> Error
done
cat Error
```

2. כתוב תכנית Script ב-Bash בשם P3.2 שקולטת רשימה של זוגות של שם תכנית ושם קובץ (שהמבנה שלה הוא כפי שמתואר בדוגמה הבאה). לכל זוג כזה של שם תכנית ושם קובץ התכנית P3.2 מפעילה את התכנית כך שכל הפלט של התכנית (גם הפלט הרגיל וגם פלט השגיאות) נכנס לקובץ ודורס את תוכנו הקודם. (אם הקובץ לא קיים אז הוא נוצר).

ניתן להניח שיש גישת הרצה לכל התכניות ברשימה. ושכל התכניות מופעלות מהתיקיה הנוכחית ממנה מופעלת התכנית P3.2. שימו לב שכאשר מפעילים את התכנית P3.2 לא מופיע פלט על המסך.

לדוגמה, נניח שלאחר הפעלת התכנית P3.2 המשתמש הקליד:

```
prog1-file1 g1-f2 prog2-file3
```

ונניח שתוכן הקבצים prog1 g1 prog2 הוא:

prog1	g1	prog2
echo hi	x=cat vv	for x in \$(seq 3)
echo1 23	y=\$(pwd)	do
echo 5555	echo 444	echo -n \$x+\$x
echo1 24		echo1 \$x
		done

אזי על המסך לא יתקבל פלט.

לאחר סיום התכנית P3.2:

תוכן הקובץ file1 הוא:

```
hi
./prog1: line 2: echo1: command not found
5555
./prog1: line 4: echo1: command not found
```

תוכן הקובץ F2 הוא:

```
./g1: line 1: vv: command not found
444
```

תוכן הקובץ file3 הוא:

```
1+1./prog2: line 4: echo1: command not found
2+2./prog2: line 4: echo1: command not found
3+3./prog2: line 4: echo1: command not found
```

פתרון שאלה 2

```
read list
for pair in $list
do
    echo $pair >| tmp
    prog=$(cut -d"-" -f1 tmp)
    file=$(cut -d"-" -f2 tmp)
    $prog >| $file 2>&1
done
```

3. כתוב תכנית Script ב-Bash בשם P3.3 שקולטת מהמשתמש רשימת שמות קבצים, כאשר כל קובץ ברשימה נמצא בתיקיה בה מופעלת התכנית P3.3 ומכיל מספרים, כאשר כל מספר בשורה נפרדת. התכנית מדפיסה לפלט שורה אחת עבור כל קובץ, שמכילה את שם הקובץ לאחר מכן תו : לאחר מכן תו רווח אחד, ולאחר מכן מספר שמציין את סכום כל המספרים שבקובץ.

לדוגמה, נניח שלאחר הפעלת התכנית P3.3 המשתמש הקליד:

```
file31 file32 g31
```

ונניח שתוכן הקבצים הוא:

```
file31      file32      g31
10           12          123
20           1
30
40
```

לדוגמה נניח שלאחר הפעלת התכנית P3.3 המשתמש הקליד:

```
file31 file32 g31
```

אזי הפלט של התכנית הוא:

```
file31: 100
file32: 13
g31: 123
```

פתרון שאלה 3

```
read list
for file in $list
do
  sum=0
  while read x
  do
    ((sum+=$x))
  done<$file
  echo "$file: $sum"
done
```

4. כתוב תכנית Script ב-Bash בשם P3.4 שמדפיסה לפלט את כל הקבצים בתיקיה הנוכחית שמתחילים באות F כאשר עבור כל קובץ מודפס בשורה נפרדת שם הקובץ, לאחריהם תו נקודותיים, לאחריהם תו רווח אחד ולאחריהם מספר שמציין את מספר התווים בקובץ (כולל תווי רווח וקפיצת שורה). בשורה האחרונה של הפלט תודפס ההודעה הבאה:

The size of the largest file starting with F is:

כשלאחר הנקודותיים יופיע תו רווח אחד ולאחריהם מספר שמציין את מספר התווים של הקובץ הגדול ביותר שמתחיל באות F, ונמצא בתיקיה הנוכחית בה מופעלת התכנית P3.4.

על סדר הקבצים בפלט להיות לפי הסדר שמתקבל לאחר הפעלת הפקודה

```
ls F*
```

בתיקיה הנוכחית.

לדוגמה, נניח שבתיקיה בה מופעלת התכנית P3.4 נמצאים הקבצים הבאים שמתחילים באות F (לפי הסדר של הפקודה ls F*).

קובץ F1 שתכנו הוא:

```
abcde ggg hh
u
z
ydfas aa
```

קובץ F123 שתכנו הוא:

```
abcd
```

קובץ FF1 שתכנו הוא:

```
123 456 789 10111212
u
z
```

לאחר הפעלת התכנית P3.4 יתקבל הפלט:

```
F1: 27
F123: 5
FF1: 25
The size of the largest file starting with F is: 27
```

פתרון שאלה 4

```
max=0
for file in $(ls F*)
do
    num=$(wc -c < $file)
    if [ $num -gt $max ]; then
        max=$num
    fi
    echo "$file: $num"
done<tmp
echo "The size of the largest file starting with F is: $max"
```