

**יטודות 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/כתבי תכנית Script ב-Bash בשם P3.1 שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

Enter program name:

והתכנית מחכה לקלט מהמשתמש שמציין שם קובץ שמכיל תכנית שנמצאת בתיקיה הנוכחית. בהמשך נקרא לקובץ זה "קובץ התכנית".

נניח לדוגמה שהמשתמש הקליד Prog1

לאחר מכן מופיעה ההודעה הבאה:

Enter list of numbers:

והתכנית מחכה לקלט מהמשתמש שמכיל סדרת מספרים (עם רווח אחד או יותר בין המספרים).

נניח למשל שהמשתמש הקליד:

```
4 2 6 1
```

לאחר מכן התכנית P3.1 יוצרת קובץ בשם temp שמכיל את כל השורות

מתוך "קובץ התכנית" שמופיעות בסדרת המספרים (לפי סדר הופעתן),  
נותנת לקובץ temp הרשאת הרצה, ומדפיסה למסך את הפלט הרגיל  
שמתקבל מהפעלת הקובץ temp ולאחר מכן מדפיסה למסך את פלט  
השגיאות שמתקבל מהפעלת הקובץ temp.

לדוגמה, נניח שתוכן "קובץ התכנית" Prog1 שנמצא בתיקיה הנוכחית  
הוא:

```
seq -s" " 8  
seq 8 -s" "  
echo abc  
seq -s"ab" 10 2 16  
for x in aa bb cc; do echo $x; done  
z=0; while [ $z -le 3 ]; do echo $z; ((z+=1)); done  
gg
```

נניח שהתכנית P3.1 קולטת מהמשתמש שם "קובץ תכנית" Prog1 וסדרת  
מספרים 1 2 6 4 כפי שתואר למעלה. אזי הפלט של התכנית P3.1  
הוא:

```
10ab12ab14ab16  
0  
1  
2  
3  
1 2 3 4 5 6 7 8  
seq: invalid floating point argument: -s  
Try `seq --help' for more information.
```

### פתרון שאלה 3.1

```
echo -n "Enter program name: "  
read filename  
echo -n "Enter list of numbers: "  
read num_list  
echo -n "" >| temp  
for num in $num_list; do  
    line_number=1  
    while read line; do  
        if [ $num -eq $line_number ]; then  
            echo "$line" >> temp  
        fi  
        ((line_number+=1))  
    done < $filename  
done  
chmod +x temp  
temp 2>|stderr.out  
cat stderr.out
```

להלן אותו פתרון בפורמט טקסט:

```
echo -n "Enter program name: "  
read filename  
echo -n "Enter list of numbers: "  
read num_list  
echo -n "" >| temp  
for num in $num_list; do  
    line_number=1  
    while read line; do  
        if [ $num -eq $line_number ]; then  
            echo "$line" >> temp  
        fi  
        ((line_number+=1))  
    done < $filename  
done  
chmod +x temp  
temp 2>|stderr.out  
cat stderr.out
```

2. כתוב/כתבי תכנית Script ב-Bash בשם P3.2 שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

Enter 4 parameters in the format prog-file-file-file:

והתכנית מחכה לקלט מהמשתמש שמכיל שם של "קובץ תכנית" ולאחריו שמות של 3 קבצים כאשר ביניהם מופיע תו אחד של -

למשל, נניח שהמשתמש מקליד: prg1-F1-F2-F3

בשלב הזה התכנית מפעילה את קובץ התכנית prg1 כאשר הקלט לתכנית הוא הקובץ F1 והפלט הרגיל מופנה לקובץ F2 ופלט השגיאות מופנה לקובץ F3.  
לאחר מכן התכנית מדפיסה שוב את ההודעה:

Enter 4 parameters in the format prog-file-file-file:

ושוב והתכנית מחכה לקלט מהמשתמש שמכיל שם של "קובץ תכנית" ולאחריו שמות של 3 קבצים כאשר ביניהם מופיע תו אחד של -

למשל, נניח שהמשתמש מקליד: prg2-G1-G2-G3

בשלב הזה התכנית מפעילה את קובץ התכנית prg2 כאשר הקלט לתכנית הוא הקובץ G1 והפלט הרגיל מופנה לקובץ G2 ופלט השגיאות מופנה לקובץ G3.

וכן הלאה, עד שהמשתמש מקליד ctrl+d

לדוגמה, נניח שתוכן הקבצים prg1 prg2 G1 F1 הוא:

prg1	prg2	F1	G1
read x	sum=0	10	40
read y	while read x	20	15
echo \$[x+y]	do	30	2
x+y	sum=\$[sum+x]		3
echo hi	done		
	echo \$sum		
	echo1 \$sum		

ונניח שהמשתמש הקליד תחילה prg1-F1-F2-F3 כפי שתואר למעלה, ולאחר מכן המשתמש הקליד prg2-G1-G2-G3 כפי שתואר למעלה, ולאחר מכן המשתמש הקליד ctrl+d

אזי על המסך לא יתקבל פלט.

לאחר סיום התכנית 3.2:

תוכן הקובץ F2 הוא:

```
30
hi
```

תוכן הקובץ F3 הוא:

```
./prg1: line 4: x+y: command not found
```

תוכן הקובץ G2 הוא:

```
60
```

תוכן הקובץ G3 הוא:

```
./prg2: line 7: echo1: command not found
```

### פתרון שאלה 3.2

```
echo -n "Enter 4 parameters in the format prog-file-file-file: "  
while read cmd; do  
    prog=$(echo $cmd | cut -d"-" -f1)  
    fn1=$(echo $cmd | cut -d"-" -f2)  
    fn2=$(echo $cmd | cut -d"-" -f3)  
    fn3=$(echo $cmd | cut -d"-" -f4)  
    $prog < $fn1 1>|$fn2 2>|$fn3  
    echo -n "Enter 4 parameters in the format prog-file-file-file: "  
done  
echo
```

להלן אותו פתרון בפורמט טקסט:

```
echo -n "Enter 4 parameters in the format prog-file-file-file: "  
while read cmd; do  
    prog=$(echo $cmd | cut -d"-" -f1)  
    fn1=$(echo $cmd | cut -d"-" -f2)  
    fn2=$(echo $cmd | cut -d"-" -f3)  
    fn3=$(echo $cmd | cut -d"-" -f4)  
    $prog < $fn1 1>|$fn2 2>|$fn3  
    echo -n "Enter 4 parameters in the format prog-file-file-file: "  
done  
echo
```

3. כתוב/כתבי תכנית Script ב-Bash בשם P3.3 שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

Enter file name:

והתכנית מחכה לקלט מהמשתמש שמכיל שם של קובץ.

למשל, נניח שהמשתמש מקליד: H1

לאחר מכן מופיעה ההודעה הבאה:

Enter 2 parameters in the format num1-num2:

למשל, נניח שהמשתמש מקליד: 2-4

לאחר מכן מופיעה שוב ההודעה:

Enter 2 parameters in the format num1-num2:

למשל נניח שהמשתמש מקליד: 1-1

לאחר מכן מופיעה שוב ההודעה:

Enter 2 parameters in the format num1-num2:

למשל, נניח שהמשתמש מקליד: 3-5

וכן הלאה עד שהמשתמש מקליד ctrl+d

בשלב הזה התכנית מדפיסה למסך את כל שורות הקובץ שמתקבלות מרשור כל העמודות שהמשתמש בחר. (הסדר הוא לפי הסדר שהמשתמש בחר ולא לפי סדר עולה של מספרי העמודות).

לדוגמה, נניח שתוכן הקובץ H1 הוא:

```
12345678
abcdefghijk
87654321
123
```

ונניח שהמשתמש הקליד תחילה H1 ולאחר מכן 2-4 ולאחר מכן 1-1 ולאחר מכן 3-5 ולאחר מכן ctrl+d אזי הפלט של התכנית למסך הוא:

```
2341345
bcdacde
7658654
2313
```

### פתרון שאלה 3.3

```
echo -n "Enter file name: "
read filename
ranges=""
echo -n "Enter 2 parameters in the format num1-num2: "
while read range; do
    ranges="$ranges $range"
    echo -n "Enter 2 parameters in the format num1-num2: "
done
echo
while read line; do
    str=""
    for range in $ranges; do
        str="$str$(echo $line | cut -c$range)"
    done
    echo $str
done < $filename
```

להלן אותו פתרון בפורמט טקסט:

```
echo -n "Enter file name: "  
read filename  
ranges=""  
echo -n "Enter 2 parameters in the format num1-num2: "  
while read range; do  
    ranges="$ranges $range"  
    echo -n "Enter 2 parameters in the format num1-num2: "  
done  
echo  
while read line; do  
    str=""  
    for range in $ranges; do  
        str="$str$(echo $line | cut -c$range)"  
    done  
    echo $str  
done < $filename
```

4. כתוב/כתבי תכנית Script ב-Bash בשם P3.4 שפועלת באופן הבא:

בהפעלת התכנית, מופיעה ההודעה הבאה:

Enter file name:

והתכנית מחכה לקלט מהמשתמש שמכיל שם של קובץ שמכיל מטריצה ריבועית של מספרים (עם רווח אחד או יותר בין המספרים בכל שורה).

למשל, נניח שהמשתמש מקליד: A1

לאחר מכן התכנית מדפיסה שורה שמתארת את סכום המספרים במסגרת החיצונית של המטריצה (כפי שמתואר בדוגמה שבהמשך).

לדוגמה נניח שתוכן הקובץ A1 הוא:

```
1 1 1 1  
2 2 2 2  
3 3 3 3  
4 4 4 4
```

ונניח שהמשתמש הקליד A1

אזי הפלט של התכנית למסך הוא:

The sum of the outside frame is: 30



הסבר לפלט:

המסגרת החיצונית של המטריצה A1 מכילה את המספרים הבאים:

```
1 1 1 1
2      2
3      3
4 4 4 4
```

והסכום שלהם הוא 30

דוגמה נוספת, אם תוכן הקובץ A2 הוא:

```
2 2 2 2 2 2
1 4 5 7 8 3
1 4 5 7 8 3
1 4 5 7 8 3
1 4 5 7 8 3
2 2 2 2 2 2
```

ונניח שהמשתמש הקליד A2

אזי הפלט של התכנית למסך הוא:

The sum of the outside frame is: 40

הסבר לפלט:

המסגרת החיצונית של המטריצה A2 מכילה את המספרים הבאים:

```
2 2 2 2 2 2
1      3
1      3
1      3
1      3
2 2 2 2 2 2
```

והסכום שלהם הוא 40

### פתרון שאלה 3.4

```
echo -n "Enter file name: "  
read filename  
top=$(echo $(head -1 $filename))  
if [ $(wc -l < $filename) -eq 1 ]; then  
    sum=$top  
else  
    bot=$(echo $(tail -1 $filename))  
    c=0  
    for x in $top; do  
        ((sum+=x))  
        ((c+=1))  
    done  
    for x in $bot; do  
        ((sum+=x))  
    done  
    i=1  
    while read line; do  
        if [ $i -ne 1 ] && [ $i -ne $c ]; then  
            left=$(echo $line | cut -d" " -f1)  
            right=$(echo $line | cut -d" " -f$c)  
            sum=$((left+right+sum))  
        fi  
        ((i+=1))  
    done < $filename  
fi  
echo "The sum of the outside frame is: $sum"
```

להלן אותו פתרון בפורמט טקסט:

```
echo -n "Enter file name: "  
read filename  
top=$(echo $(head -1 $filename))  
if [ $(wc -l < $filename) -eq 1 ]; then  
    sum=$top  
else  
    bot=$(echo $(tail -1 $filename))  
    c=0  
    for x in $top; do  
        ((sum+=x))  
        ((c+=1))  
    done  
    for x in $bot; do  
        ((sum+=x))  
    done  
    i=1  
    while read line; do  
        if [ $i -ne 1 ] && [ $i -ne $c ]; then  
            left=$(echo $line | cut -d" " -f1)  
            right=$(echo $line | cut -d" " -f$c)  
            sum=$((left+right+sum))  
        fi  
        ((i+=1))  
    done < $filename  
fi  
echo "The sum of the outside frame is: $sum"
```