

יסודות מערכות פתוחות
תרגיל מס' 3

מועד ההגשה: 1.12.2014

שימו לב: התכניות עוברות בדיקה אוטומטית ולכן על הפלט של התכניות להיות בדיוק כפי שנדרש ללא תוספות, (אין להוסיף הסברים, רווחים מיותרים, שורות ריקות מיותרות וכו').

בכל השאלות אנחנו קוראים תיקיה נוכחית לתיקיה בה מופעלת התכנית.

בכל השאלות ברגע שמתייחסים לקבצים/תיקיות בתיקיה הנוכחית אסור להשתמש בשם המלא של הקבצים/תיקיות (אלא צריך להשתמש בשם היחסי שלהם). הסיבה לכך היא שתכנית הבדיקה האוטומטית מריצה את התכניות שלכם בחשבון הקורס. השם המלא של התיקיה הנוכחית בחשבון הקורס שונה מהשם המלא של התיקיה הנוכחית בחשבון שלכם ולכן אסור להשתמש בשם המלא, אלא צריך להשתמש בשם היחסי.

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 |>

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 שמפעילה תכנית בשם GG שנמצאת בתיקיה הנוכחית, כך שעל המסך יוצג הפלט הרגיל של התכנית GG, לאחריו מספר שמציין את מספר השורות בפלט הרגיל של התכנית, לאחריו פלט השגיאות של התכנית, ולאחריו מספר שמציין את מספר השורות בפלט השגיאות של התכנית. (התוכנית P3.1 לא יוצרת את התוכנית GG אלא מניחה שהיא קיימת).

לדוגמה:

נניח שתוכן התכנית GG שנמצאת בתיקיה הנוכחית הוא:

```
seq 5
pwd
echo1 abc
seq 100 100 400
echo2
echo 3
seq10
echo 8
x=10
echo $x
echo $x >| hh
```

לאחר הפעלת התכנית P3.1 יתקבל הפלט הבא:

```
1
2
3
4
5
/home/cs/segel/basicsys/tt
100
200
300
400
3
8
10
13
./GG: line 3: echo1: command not found
./GG: line 5: echo2: command not found
./GG: line 7: seq10: command not found
3
```

.2

כתוב תכנית Script ב-Bash בשם P3.2 שקולטת מהמשתמש שתי מחרוזות. המחרוזת הראשונה שהמשתמש מקליד מציינת שם קובץ שמכיל תכנית (בהמשך נקרא לה תכנית 1) והמחרוזת השניה שהמשתמש מקליד מציינת שם קובץ (בהמשך נקרא לו קובץ 1). התכנית P3.2 מפעילה את תכנית 1 כך שכל הפלט של תכנית 1 (גם הפלט הרגיל וגם פלט השגיאות) יכנס לקובץ 1 וידרוס את תוכנו.

לדוגמה:

נניח שתוכן הקובץ BB שנמצא בתיקיה הנוכחית הוא:

```
seq 10 20 50
echo1 abc
echo xyz
pwd1
echo def
```

ונניח שלאחר הפעלת התכנית P3.2 המשתמש יקליד BB במחרוזת הראשונה ו-F3 במחרוזת השניה. התכנית P3.2 תסתיים ללא הדפסת פלט למסך, ותוכן הקובץ F3 ישתנה ויהיה:

```
10
30
50
./BB: line 2: echo1: command not found
xyz
./BB: line 4: pwd1: command not found
def
```

.3

כתוב תכנית Script ב-Bash בשם P3.3 שקולטת מהמשתמש שתי מחרוזות לראשונה נקרא בהמשך מחרוזת 1 ולשניה נקרא בהמשך מחרוזת 2. התכנית P3.3 מדפיסה לפלט שתי שורות: בשורה הראשונה מספר שמציין את מספר התווים במחרוזת 1 ובשורה השניה מספר שמציין את מספר התווים במחרוזת 2.

לדוגמה, נניח שלאחר הפעלת התכנית P3.3 המשתמש הקליד את המחרוזת abcd ולאחר מכן המשתמש הקליד את המחרוזת xyz אז הפלט של התכנית יהיה:

```
4
3
```

.4

כתוב תכנית Script ב-Bash בשם P3.4 שקולטת מהמשתמש מספרים (כל מספר בשורה נפרדת) עד שהמשתמש מקליד ctrl+d ומדפיסה לפלט שורה אחת שמכילה את המספר הקטן ביותר מבין המספרים שהמשתמש הקליד. (גם אם המספר הקטן מופיע יותר מפעם אחת ברשימת המספרים שהמשתמש מקליד, בפלט של התכנית הוא יופיע רק פעם אחת בלבד).

לדוגמה, נניח שלאחר הפעלת התכנית P3.4 המשתמש הקליד את סדרת המספרים הבאה ולאחריה הקליד ctrl+d :

```
10
-20
551
3
-20
5
```

אז התכנית תדפיס למסך שורה אחת בדיוק שמכילה:

```
-20
```

עליכם להגיש את כל קבצי ה- script דהינו הקבצים: P3.1 - P3.4
לבדיקה אוטומטית ע"י הפקודה:

~basicsys/submit

יש להריץ את הפקודה הנ"ל מהתיקיה שלכם בה נמצאים הקבצים
P3.1 - P3.4 ניתן להשתמש בפקודה הנ"ל להגשת קבצי ה- script
פעם אחת בלבד. אם ברצונכם להגיש את הקבצים פעם נוספת עליכם
לפנות למרצה הקורס. מותר להגיש בזוגות אך עדיף להגיש לבד.

בהצלחה !