

מבוא לתכנות מערכות

תרגיל בית מספר 1 (Bash)

סמסטר אביב 2017

תאריך פרסום: 2.4.2017
תאריך הגשה: 23.4.2017, 23:55
משקל התרגיל: 5% מהציון הסופי (תקף)
מתרגל אחראי: אילון שושן (eylon.mtm@gmail.com)

1 הערות כלליות

- מותר (ומומלץ) להגיש בזוגות.
- שימו לב: לא יינתנו דחיות למועד הגשת התרגיל. תכננו את הזמן בהתאם.
- קראו את המסמך "[מידע כללי לגבי תרגילי הבית](#)".
- לשאלות בנוגע לתרגיל יש לפנות במייל למתרגל האחראי על התרגיל. נא לכתוב בשורת הנושא (subject): mtm1. לפני שליחת שאלה - נא וודאו שהיא לא נענתה כבר ב-F.A.Q ושהתשובה אינה ברורה ממסמך זה, מהדוגמא או מהבדיקות שפורסמו עם התרגיל.
- מומלץ להגיע לסדנאות לקבלת תמיכה בתרגיל (אימיילים לגבי בעיות במימוש שלכם לא יענו).
- קראו מסמך זה עד סופו ועיברו על הדוגמא שפורסמה לפני תחילת הפתרון.
- חובה להתעדכן [בעמוד ה-F.A.Q של התרגיל](#).
- העתקות קוד בין סטודנטים יטופלו בחומרה!

2 התרגיל

2.1 רקע

התרגיל עוסק בכתיבת סקריפטים של bash שמטרתם ניהול database המייצג את רשת חדרי הבריחה MTM-ESCAPY. ה-database שלנו מיוצג ע"י עץ של קבצים שנתאר מיד. עץ הקבצים הנ"ל נמצא בתוך תיקייה אשר נכנה "התיקייה הראשית".

2.2 הקבצים שמסופקים לכם

- חוץ ממסמך זה, אתם מקבלים עבור התרגיל את הקבצים הבאים:
1. קובץ בשם MainDirectoryExample.zip אשר מכיל תיקייה בשם MTMEscapy, שהיא דוגמא ל-"תיקייה ראשית".
 2. קובץ בשם SampleTests.zip - אשר מכיל 12 בדיקות לדוגמא. כל בדיקה היא למעשה שני קבצים: קובץ הרצה (לדוגמא test1) וקובץ פלט צפוי (לדוגמא test1.expout). קובץ ההרצה מפעיל את אחד מהסקריפטים שהייתם אמורים לכתוב. אם הסקריפטים שלכם עובדים כמו שצריך - הפלט של קובץ ההרצה (ל-stdout) יהיה זהה לקובץ הפלט הצפוי המתאים, וכמו כן לא יהיה פלט ל-stderr כלל¹. כל הבדיקות הנ"ל הן יופעלו מהתיקייה הראשית שב-MainDirectoryExample.zip.

¹ אם יש פלט ל-stderr, כנראה Bash נתקלה בפקודה עם סינטקס לא חוקי, ואז Bash מדפיסה הודעת שגיאה ל-stderr, או לחילופין הפעלתם תוכנית סטנדרטית (כמו cut) עם ארגומנטים לא חוקיים, ואז אותה תוכנית מדפיסה ל-stderr.

2.3 הגדרת עץ הקבצים

נתאר כעת בפירוט את מבנה התיקייה הראשית. כאמור בקובץ MainDirectoryExample.zip שסופק לכם יש דוגמא ל"תיקייה ראשית" (שם התיקייה הוא MTMEscapy).

התיקייה הראשית מכילה את כל המידע על המבקרים ב-MTM-ESCAPY. בתוך התיקייה הראשית, עבור כל חדר בריחה ברשת, קיימת תיקייה ששמה הוא שם הסניף של אותו חדר בריחה. שם חדר הבריחה מכיל אותיות קטנות באנגלית בלבד (אחת או יותר). כל תיקייה כזו (שמייצגת חדר בריחה) מכילה את הקבצים הבאים:

1. קובץ בשם visitors.txt שכל שורה בו מייצגת ביקור כלשהו בחדר הבריחה, והינה בפורמט הבא:

<Level> <Visit ID> <Visitor>

(כאן הסימנים < ו- > הינם להבהרה בלבד, בפועל הם אינם מופיעים בקובץ – ראו דוגמאות קלט מסופקות) כאשר:

- **Level** – רמת הקושי המבוקשת בחדר הבריחה – מספר שלם חיובי.
- **Visit ID** – מזהה ביקור בחדר הבריחה – מספר שלם חיובי.
 - שימו לב: מזהה הביקור הינו ייחודי עבור כל ביקור בחדר הבריחה. כלומר לא ייתכנו שני ביקורים בחדר בריחה מסוים, עם אותו מזהה.
- **Visitor** – שם המבקר בחדר הבריחה.
 - שימו לב:
 - שם המבקר מורכב **ממילה אחת בלבד**, המורכבת מאותיות באנגלית בלבד וללא רווחים.
 - שם המבקר משמש בתור מזהה ייחודי. כלומר, אם מופיע משתמש בשם X בחדר בריחה מסויים, ובחדר בריחה אחר מופיע ביקור של אדם בשם X, אזי מדובר באותו מבקר.

2. תיקייה בשם challenges/ המכילה קבצים עם השמות chg.<challenge ID>.

- מבקרים בחדרי הבריחה השונים יכולים (אך לא חייבים) לבחור אתגר מסוג כלשהו לבחירתם בחדר הבריחה.
- מבקר יכול לבחור **לכל היותר אתגר אחד** בביקור כלשהו בחדר הבריחה.
 - ניתן להניח שבמערכת הקבצים המסופקים, נלקח לכל היותר אתגר אחד בכל ביקור בחדר בריחה.
- אם מבקר יבחר על עצמו אתגר בחדר בריחה X, אז יישמר עבורו קובץ chg. מתאים בתיקייה X/challenges.

קבצי chg. יהיו במבנה הבא:

visit id: <Visit ID>	מזהה ביקור בחדר הבריחה (מספר שלם חיובי)
overall time: <Time To Exit>	משך הזמן (בדקות) מתחילה סיום (מספר שלם חיובי)
challenge type: <Challenge Type ID>	מזהה סוג האתגר שנלקח (מספר שלם חיובי)
challenge description: <Description>	תיאור מילולי של האתגר שנלקח

שימו לב:

- התיקייה הראשית יכולה להכיל קבצים בנוסף לתיקיות חדרי הבריחה. אך ניתן להניח שתיקיות בהכרח מייצגות חדרי בריחה.
- תיקיית ה-challenges יכולה להכיל קבצים נוספים שאינם בסיומת chg או תיקיות, שאינם בפורמט chg המוכר (שפורט לעיל).

2.4 הסקריפטים שעליכם לכתוב

עליכם לכתוב את 5 הסקריפטים שנתאר מיד. כל סקריפט מוגדר ע"י אופן הקריאה שלו (הארגומנטים שהוא מקבל), והפלט שהוא נדרש להדפיס לערוץ הפלט הסטנדרטי (stdout). כל 5 הסקריפטים הללו אינם מקבלים קלט דרך ערוץ הקלט הסטנדרטי (stdin) ואינם מדפיסים לערוץ השגיאה הסטנדרטי (stderr). כמובן שסקריפטי העזר שלכם יכולים לקבל קלט דרך ערוץ הקלט הסטנדרטי.

שימו לב שלכל אחד מ-5 הסקריפטים - סיפקנו לכם בדיקות לדוגמא (סך הכל קיבלתם 12 בדיקות).

אלא אם נאמר בפירוש אחרת, ניתן להניח שהארגומנטים שאיתם מפעילים את הסקריפטים תמיד חוקיים (כלומר לא חסרים ארגומנטים, אין ארגומנטים מיותרים, וכל ארגומנט הוא מחרוזת חוקית עבור שם חדר בריחה/מספר שלם/שם מבקר - בהתאם למשמעות הארגומנט).

כל הסקריפטים שאתם כותבים (כולל סקריפטי העזר) חייבים להימצא באותה תיקייה, אשר חייבת להיות מחוץ ל"תיקייה הראשית". מיקום תיקיית הסקריפטים אינו משנה, אך תדאגו להוסיף את המסלול (path) של אותה תיקייה למשתנה PATH² כפי שלמדנו בתרגול 1 (אתם יכולים להניח שגם אנחנו עושים זאת בזמן בדיקת הסקריפטים שלכם).

נאשר מריצים את הסקריפטים שלכם, התיקייה הנוכחית היא ה"תיקייה הראשית" הרלוונטית (שעבורה הסקריפטים אמורים לעבוד).

כתבו את הסקריפטים הבאים :

2.4.1 last_visitors

אופן הרצה :

```
last_visitors <ROOM_NAME> [<VISITORS_AMOUNT>]
```

תיאור

מוצא את VISITORS_AMOUNT המבקרים האחרונים בחדר הבריחה ששמו ROOM_NAME מתוך הקובץ visitors.txt ומדפיס אותם.

VISITORS_AMOUNT הוא ארגומנט אופציונלי. אם לא הוגדר מספר ב-VISITORS_AMOUNT, או אם ישנם פחות ביקורים בקובץ visitors.txt מאשר VISITORS_AMOUNT, יש להדפיס את כל שמות המבקרים שמופיעים בקובץ.

הפלט :

כל שורה בפלט היא שם (ייחודי) של מבקר.

סדר שורות הפלט : על השורות להיות ממויינות לפי מזהה הביקור של הביקור האחרון עבור כל מבקר, בסדר יורד. כלומר, מבקר בחדר הבריחה שביקר יותר לאחרונה, הוא בעל מזהה ביקור גדול יותר ולכן יופיע מוקדם יותר בשורות הפלט.

שימו לב :

- שם של מבקר צריך להופיע פעם אחת בשורות הפלט, בהתאם לביקור האחרון שלו בחדר הבריחה.
- אין הגבלה על מספר המבקרים השונים בחדר הבריחה.

2.4.2 visitor_for_room

אופן הרצה :

```
visitor_for_room <ROOM_NAME> <VISITOR>
```

תיאור

מוצא את כל הביקורים של VISITOR בחדר הבריחה ROOM_NAME מתוך הקובץ visitors.txt.

הפלט :

כל שורה בפלט מייצגת ביקור של VISITOR בחדר הבריחה ROOM_NAME ותהיה בעלת המבנה הבא :

² לדוגמא ע"י הפקודה הבאה (שתפעילו ב-Bash לפני השימוש בסקריפטים) :

```
PATH="$PATH:~/some/path/to/my-scripts"
```

<Level> <Visit ID>

כלומר, עבור כל ביקור של אותו אדם בחדר הבריחה, תיפלט שורה ובה רמת הקושי באותו ביקור, ומזהה הביקור המתאים.
סדר שורות הפלט: לפי מזהה הביקור Visit ID **בסדר עולה** (בניגוד לסקריפט הקודם).

שימו לב:

יש לבצע חיפוש של שם המבקר במדויק. כך למשל, Ben ו-Beni הם שמות שונים. חפשו בתרגולים את הדגל המתאים.

2.4.3 visitor_challenges

אופן הרצה:

visitor_challenges <ROOM_NAME> <VISITOR> <THRESHOLD>

תיאור

מדפיס את כל האתגרים שלקח על עצמו המבקר VISITOR בחדר הבריחה ROOM_NAME.

הפלט:

כל שורה מייצגת אתגר שלקח על עצמו VISITOR בחדר הבריחה ROOM_NAME ותהיה בעלת המבנה הבא:

<Visit ID> <Title>

כאשר:

- Visit ID הוא מזהה הביקור שבו לקח על עצמו VISITOR את האתגר.
- Title יהיה אחת מהאפשרויות **Winner** או **Loser**, בהתאם למידת הצלחתו של המבקר באותו אתגר.
- אם **משך הזמן** שלקח ל-VISITOR לצאת מחדר הבריחה **קטן ממש** מ-Threshold, אז הוא יוגדר כ-Winner. אחרת, יוגדר כ-Loser.

סדר שורות הפלט: לפי מזהה הביקור Visit ID **בסדר עולה**.

רמז: מזהה הביקור (Visit ID) מקשר בין שם המבקר ב-visitors.txt לבין האתגר בקובץ עם סיומת .chg.

דוגמה: כך למשך, אם John ביקר פעמיים בחדר הבריחה, ובכל אחת מהפעמים לקח על עצמו אתגר: את הביקור שהמזהה שלו 10 סיים ב-50 דקות, ואת הביקור שהמזהה שלו הוא 11 סיים ב-40 דקות.

עבור הרצה שבה: Visitor=John, Threshold=45, יודפס:

10 Loser
11 Winner

שימו לב:

- Threshold הוא סף זמן (בדקות) – מספר שלם חיובי.

2.4.4 room_challenges

אופן הרצה:

room_challenges <ROOM_NAME> <THRESHOLD>

תיאור

עבור כל מבקר בחדר הבריחה ROOM_NAME מדפיס את שמו, מספר האתגרים אותם לקח בחדר הבריחה וקיבל תואר Winner, ואת מספר האתגרים אותם לקח בחדר הבריחה וקיבל תואר Loser.

הפלט:

כל שורת פלט תהיה בעלת המבנה הבא:

<VISITOR> <#Winner> <#Loser>

כאשר:

- VISITOR הוא שם המבקר בחדר הבריחה.
- #Winner הוא מספר האתגרים אותם לקח וקיבל תואר Winner.
- #Loser הוא מספר האתגרים אותם לקח וקיבל תואר Loser.

סדר שורות הפלט: במיון ראשי על פי **כמות האתגרים הכוללת** שלקח כל מבקר בסדר יורד, ובמיון משני **בסדר עולה על פי סדר לקסיקוגרפי** של שמות המבקרים.

דוגמה:

אם:

- Ziv לקח על עצמו 5 אתגרים, ב-3 קיבל תואר Winner וב-2 קיבל תואר Loser.
- Shir לקחה על עצמה 4 אתגרים, וקיבלה תואר Winner בכל הארבעה.
- Eli לקח על עצמו 4 אתגרים, וקיבלה 2 תארי Winner ו-2 תארי Loser.

הפלט יהיה:

Ziv 3 2
Eli 2 2
Shir 4 0

Ziv מופיע ראשון כי לקח על עצמו את המספר הגבוה ביותר של אתגרים (5 בסך הכול).
Eli ו-Shir התנסו שניהם ב-4 אתגרים, ולכן סדר ההדפסה ביניהם ייקבע על פי הסדר הלקסיקוגרפי של שמותם.
מכיוון שכך, Eli מופיע לפני Shir.

שימו לב:

- קבלת תואר Winner/Loser מוגדרת כמו בסקריפט הקודם, ונקבעת בהתאם לערכו של THRESHOLD.

2.4.5 mtm_escapy_challenges

אופן הרצה:

mtm_escapy_challenges <THRESHOLD> [<VISITOR_1> <VISITOR_2> ... <VISITOR_N>]

תיאור

עבור כל מבקר שצויין ברשימת הארגומנטים, מדפיס את תוצאות האתגרים שלקח בכל חדר בריחה ברשת MTM-ESCAPY.

הפלט:

על פי סדר השמות שהוכנסו כארגומנטים עבור כל מבקר תודפס שורה המכילה את שמו.

לאחר מכן, עבור כל חדר בריחה בשם ROOM_NAME בו ביקר יודפסו עבור מבקר זה, שורות סיכום **המתחילות ב- TAB** ובמבנה הבא:

<ROOM_NAME>: <#Winner> <#Loser>

כאשר:

- #Winner הוא מספר האתגרים אותם בחר המבקר וקיבל תואר Winner.
- #Loser הוא מספר האתגרים אותם בחר המבקר וקיבל תואר Loser.

סדר שורות הפלט: הפלט עבור המשתמשים השונים יופיע כסדר הארגומנטים שבו הוכנסו בשורת הרצת הסקריפט. בפלט הספציפי של כל מבקר, יש להדפיס **בסדר עולה** לקסיקוגרפית של שמות חדרי הבריחה.

שימו לב:

- ייתכן כי התיקייה הראשית ריקה/איננה מכילה תיקיות המייצגות חדרים. במקרה כזה, הפלט עבור כל מבקר צריך להיות ריק.
- כאמור, התיקייה הראשית יכולה להכיל קבצים שאינם תיקיות, ועליכם להתעלם מהם.
- קבלת תואר Winner/Loser מוגדרת כמו בסקריפט הקודם, ונקבעת בהתאם לערכו של THRESHOLD.
- כאמור, כל שורת סיכום תתחיל ב-TAB לשם הזחה, במטרה להשיג ויזואליות טובה של הפלט המודפס.
- אם מבקר לא ביקר בחדר בריחה כלשהו מתוך הרשת MTM-ESCAPY, אין לכלול שורת פלט עם חדר בריחה זה עבורו.
- o בפרט, אם לא נמצא אף ביקור עבור שם מבוקש, הפלט עבורו יהיה ריק. כך למשל, בדוגמה המוצגת בהמשך, הפלט עבור השם המבוקש Moti הינו ריק.

דוגמה:

אם נניח שעבור Threshold=45 מתקיים:

- Roi לקח על עצמו 3 אתגרים בחדר הבריחה escapeisis. ב-1 קיבל תואר Winner וב-2 הנותרים קיבל תואר Loser.
- Roi ביקר בחדר הבריחה mentality אך לא לקח על עצמו אתגרים שם כלל.
- Noam ביקר בחדר הבריחה escapeisis, לקחה על עצמה אתגר אחד ובו קיבלה תואר Loser.
- Noam לא ביקרה באף חדר חדר בריחה אחר, מלבד escapeisis (בפרט לא ביקרה ב-mentality).
- Moti מעודו לא ביקר באף חדר בריחה ברשת MTM-ESCAPY.

עבור ההרצה הבאה של הסקריפט:

mtm_escapy_challenges 45 Roi Noam Moti

נקבל כי פלט הסקריפט יהיה:

```
Roi
    escapeisis: 1 2
    mentality: 0 0
Noam
    escapeisis: 0 1
Moti
```

3 הגשה

- את ההגשה יש לבצע דרך אתר הקורס, תחת Electronic Submit <- HW1 <- Assignments. הקפידו על הדברים הבאים:
- יש להגיש קובץ ZIP בודד (לא RAR או כל דבר אחר) בשם hw1.zip, אשר מכיל אך ורק את הסקריפטים (כולל סקריפטי העזר) שכתבתם. כל הסקריפטים שלכם חייבים להופיע בתיקיית השורש בתוך קובץ ה-ZIP. כלומר, כאשר התיקייה הנוכחית היא התיקייה שמכילה את הסקריפטים שלכם, בצעו את הפקודה הבאה ליצירת קובץ ההגשה:
> zip hw1.zip *
 - השמות של 5 הסקריפטים שנדרשתם לכתוב חייבים להיות זהים לשמם כפי שהופיעו במסמך זה (זכרו שמות הקבצים הם case sensitive במערכות הפעלה דמויות Unix, ושטעות קטנה בשם של סקריפט תכשיל בדיקה).
 - **חובה לשמור את קוד האישור אשר מתקבל בעת ההגשה, וחובה לשלוח אותו לשותף.** אם לא תעשו זאת - אתם עלולים לקבל 0 בתרגיל על שטות (הגשה לקורס/תרגיל הלא נכון).
 - שימו לב, ניתן להגיש את התרגיל מס' פעמים. ההגשה האחרונה היא הנחשבת.
 - שימו לב שאתם מגישים למקום הנכון - לא לאתר של קורס אחר, ולא כפתרון של תרגיל בית אחר.
 - **חובה להריץ את סקריפט הבדיקה הסופית על הקובץ שמגישים! ראו סעיף 4.**

4 סקריפט הבדיקה הסופית

על מנת לוודא שקובץ ההגשה שלכם תקין, לפני שאתם מגישים, עליכם להריץ דרך חשבון ה-CSL2 שלכם סקריפט ייעודי שהכנו. הסקריפט מוודא שמבנה קובץ ה-zip שאתם מגישים תקין, ושהסקריפטים שאתם מגישים אכן עוברים את הבדיקות שפורסמו עם התרגיל. את סקריפט הבדיקה מריצים ע"י הפקודה הבאה (על ה-CSL2):
~mtm/public/1617a/ex1/final_check <submission file>
כאשר <submission file> הוא קובץ ה-zip שאתם מעוניינים להגיש.

לדוגמא, כך נראית הרצה מוצלחת של סקריפט הבדיקה:

```
>~mtmchk/public/1617b/ex1/final_check hw1.zip  
This script will now run your scripts with the published tests.  
Continue? (y/n)y  
Running test 1: success  
Running test 2: success  
Running test 3: success  
Running test 4: success  
Running test 5: success  
Running test 6: success  
Running test 7: success  
Running test 8: success  
Running test 9: success  
Running test 10: success  
Running test 11: success  
Running test 12: success  
Everything went well :)  
Final check passed.
```

5 דרישות, הגבלות והערות כלליות

- תו ירידת השורה שונה בין windows לunix לכן כאשר מעתיקים קבצים מ- windows ל- linux יש לבצע את הפקודה
dos2unix <file name>
דוגמא:
dos2unix a.txt b.txt
Convert and replace a.txt. and b.txt.
- **אסור להשתמש בקבצים זמניים.**
- יש לכתוב את התרגיל ב-Bash בלבד (ולא בשפות script אחרות או ב-C/C++). קראו את ההסבר על "החלפת ה-shell" במסמך "[מידע כללי לגבי תרגילי הבית](#)".
- **אסור להשתמש בתוכניות Unix סטנדרטיות אשר לא נלמדו בתרגולים, וגם לא בדגלים שלא נלמדו.**
- היעזרו בתוכניות הסטנדרטיות שנלמדו (uniq, cut, וכו'). שימוש בפקודות אלה יכול לקצר בהרבה את כתיבת הסקריפט!
- הסקריפטים ייבדקו על CSL2.
- העובדה שהבדיקה תבצע על CSL2 אינה אומרת שאתם חייבים כל הזמן לפתח על CSL2. אתם יכולים לפתח את הסקריפטים שלכם במחשבים אחרים ובהמשך להעביר אותם ל-CSL2 ולוודא שהם עובדים נכון על ה-CSL2. **שימו לב –** סקריפטים אשר עובדים נכון במחשב אחר עלולים ליצור פלט שגוי על CSL2 – לפני ההגשה יש לוודא שהסקריפטים שלכם עובדים נכון על ה-CSL2!
- אפשר לפתח את הסקריפטים על מחשבי הלינוקס בחווה, או לחילופין ב-Cygwin שעל מחשבי Windows בחווה, או להתקין Cygwin במחשב Windows שלכם ([ראו מדריך](#)).
- **שימו לב:** בד"כ יש עומס רב על CSL2 בימים האחרונים לפני הגשת תרגיל. מומלץ מאוד לא לדחות את התרגיל ליומיים האחרונים. מומלץ גם לפתח את הסקריפטים במחשב אחר כפי שהוסבר לעיל בכדי לצמצם חשיפה לבעיות עומס/איטיות של CSL2. צוות הקורס לא יוכל לטפל בבעיות שקשורות לעומס/איטיות ב-CSL2. לא יינתנו שום הקלות/דחיות בגלל העומס/איטיות של CSL2.
- ה-shell שצריך להריץ את התוכניות הוא /bin/bash
כלומר השורה הראשונה בכל סקריפט תהיה:
#!/bin/bash
- הניחו שבזמן הבדיקה שלנו - אנו נחלץ את קבצי הסקריפט שלכם לתיקייה כלשהי, ונוסיף את המסלול (path) של התיקייה למשתנה PATH. כלומר על מנת להריץ סקריפט עזר, יש לכתוב את שם סקריפט העזר ללא מסלול כלשהו.
- אסור להריץ סקריפטים ע"י הפקודה source.
- אין צורך לכתוב את הסקריפטים לפי מוסכמות קוד (code conventions) כלשהן.
- לא חייבים לתעד את הסקריפטים. מומלץ בחום לציין בראש כל סקריפט בהערה תמציתית את אופן הפעלת הסקריפט - והסבר תמציתי לגבי מה הוא עושה. כנ"ל לגבי פונקציות עזר אם אתם משתמשים בהן.
- הקפידו לא להדפיס דברים מיותרים לערוצי הפלט - **טעות קלה של אפילו תו רווח יחיד תכשיל בדיקה שלמה.**
- הנכם רשאים להניח כי לא קיימות שגיאות בתיקייה הראשית (במבנה עץ הקבצים ובתוכן הקבצים).
- בזמן תכנון הסקריפטים שלכם - השתדלו כמה שיותר לעשות שימוש חוזר בקוד קיים במקום לכתוב קוד חדש. עיקרון זה (reusability) הינו בעל חשיבות אדירה בקורס זה ובכלל בחיים, והוא לא רק יחסוך לכם כתיבת קוד, אלא גם זמן דיבוג ותחזוקה (במקרה הנפוץ של תוכנה שמשתמשים בה לאורך זמן).
- כל קבצי הטקסט המוזכרים בתרגיל וכן כל הפלטים הנדרשים מהסקריפטים שלכם מקודדים ב-ASCII, ו-newline (אשר מסיים כל שורה, כולל השורה האחרונה) נעשה ע"י התו '\n', כנהוג בסביבת unix.
- השתדלו לכתוב סקריפטים קצרים ככל האפשר, ולא דווקא יעילים ככל האפשר. כאמור - הציון נקבע רק על סמך היכולת של הסקריפטים שלכם לעבור בדיקות (כמו ה-12 בדיקות לדוגמא שניתנו לכם).
- אין דרישה על ה-exit status של הסקריפטים.

6 שינויים עדכונים והודעות בנוגע לתרגיל

- הודעות דחופות בנוגע לתרגיל יפורסמו בעמוד ההודעות הראשי של אתר הקורס:
<http://webcourse.cs.technion.ac.il/234122/Spring2017/news.html>
חובה להתעדכן [F.A.Q של התרגיל](#) (בדף זה יפורסמו שאלות ותשובות נפוצות בנוגע לתרגיל).

בהצלחה !