

# מבוא לתכנות מערכות - 234122

## תרגיל בית מספר 3 (Bash)

### סמסטר אביב תשע"ו

תאריך פרסום: יום שלישי 10.5.2016  
מועד אחרון להגשה: יום שלישי 24.5.2016 23:55  
משקל התרגיל: 5% מהציון הסופי (תקף)  
מתרגל אחראי: ישראל גוטר [gisrael@cs.technion.ac.il](mailto:gisrael@cs.technion.ac.il)

#### 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", ובפרט עקבו אחר ההוראות בסעיף 1.2 במסמך זה על מנת להגדיר את Bash בתור shell ברירת המחדל על השרת T2.
- 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 בעמוד ה-F.A.Q ושהתשובה אינה ברורה ממסמך זה ומהדוגמאות שפורסמו עם התרגיל.
- מומלץ מאוד להגיע לסדנאות לקבלת תמיכה בתרגיל (לא נוכל לטפל באימיילים לגבי בעיות במימוש שלכם).
- קראו מסמך זה עד סופו לפני תחילת הפתרון.
- העתקות קוד בין סטודנטים יטופלו בחומרה!
- שינויים שנעשו במסמך זה מאז פרסומו מודגשים בצהוב.

## 2 התרגיל

### 2.1 רקע

סוכנות "כח לשוכר" הוקמה במטרה לסייע לשוכר הנבון למצוא דירה להשכרה כפי מבוקשו. הקושי הוא בריבוי של סוכנויות תיווך קיימות אשר מפרסמות, כל אחת, רשימות של דירות להשכרה באופן אסינכרוני. סוכנות "כח לשוכר" מעוניינת להעמיד לרשות השוכרים אפליקציה אשר תקל עליהם את מלאכת בחינת הרשימות השונות העומדות לרשותם מהסוכנויות השונות הקיימות. הסוכנות פנתה לפקולטה למדעי המחשב בטכניון וביקשה את עזרתה. לאחר שהנושא נדון בדרגים בכירים, הוחלט שיהיה נוח להעמיד לרשות הסוכנות תוכנית script ראשית ב-Bash. הפקולטה פנתה לצוות קורס מת"ם בהקשר לעניין זה. צוות הקורס התלהב ביותר והחליט להטיל את המלאכה על הסטודנטים בקורס, מתוך ביטחון שהסטודנטים יהיו מלאי עניין ומוטיבציה להעמיד לרשות "כח לשוכר" תוכנית כזו. בכך הם, למעשה, יעזרו גם לעצמם, אם הם, בעצמם, מבקשים לשכור דירה. אגב - הפקולטה לא פרסמה את הסיבות להחלטה לפתח תוכנית script ראשית ב-Bash מתוך הנחה שהגורמים המתאימים יבינו בעצמם את השיקולים שהביאו להחלטה זו. צוות הקורס מת"ם הסכים עם הפקולטה, והשאיר לסטודנטים להבין בעצמם מדוע קל ונוח לפתח תוכנית כזו ב-Bash. (האמנם ?)

### 2.2 המשימה

בסעיף זה מוגדרת ומתוארת המשימה הראשית. בסעיף 2.5 מפורטות הנחיות נוספות באשר למימוש המשימה כפי שתואר.

נדרש לכתוב תוכנית script ראשית ששמה `getApartments`. הפעלת התוכנית תבוצע משורת הפקודה (command line) כדלקמן:

```
getApartments <file> <city> <limit>
```

התוכנית מקבלת בשורת הפקודה 3 פרמטרים כדלקמן :

<file> זהו שם של קובץ טקסט אשר מכיל בתוכו אוסף מקורות שמגיעים מסוכנויות התיווך השונות. מפאת אי ההתאמה בין הסוכנויות השונות, חלק מהסוכנויות מפרסמות את הרשימות בקבצים בודדים וחלק מפרסמות ב – `directories`. ייתכנו נתונים כפולים שמופיעים במספר סוכנויות. קבצים בעלי אותו שם אשר מגיעים מסוכנויות שונות אינם בהכרח זהים. הקובץ <file> מכיל שמות של קבצים ו/או שמות של `directories`. כל שם כזה מופיע בשורה נפרדת. תיתכנה גם שורות ריקות.

תיתכנה גם הערות. הערה מתחילה בתו # ונמשכת ממנו ועד סוף השורה. מובטח שהתו # לא מופיע באמצע מילה, אבל הוא יכול להופיע צמוד לתחילת מילה שנמצאת מימין לו. מובטח שהתוכן של הקובץ תקין (אם הקובץ קיים).

השמות שרלוונטיים לענייננו הם שמות שמסתיימים ב - `flat`. זה רלוונטי גם לשמות של קבצים וגם לשמות של `directories`. יש להתעלם מכל דבר אחר שיכול, גם הוא, להופיע ברשימה. `directory` יכול להכיל בתוכו קבצים אבל גם `directories` נוספים אשר, גם הם, יכולים להכיל בתוכם קבצים או `directories` נוספים וכך הלאה. חיפוש הדירות להשכרה יבוצע בכל הקבצים הרלוונטיים אשר שמם מסתיים, כאמור, ב - `flat`. הערה – שם של קובץ או `directory` יכול להכיל גם `path`. אם אינו מכיל `path` אזי מתכוונים ל-`directory` הנוכחי שבו נמצאים.

<city> זהו שם של עיר שבה מחפשים את הדירה להשכרה. שם זה יכול להכיל מילה אחת וגם מספר מילים. במקרה שבו ישנן מספר מילים מופיעים רווחים בין המילים (אחד או יותר). שם עיר יכול להינתן באותיות גדולות ו/או קטנות, חלקן או כולן. ההתייחסות לשם עיר תבוצע תמיד תוך התעלמות מהבדלים בין אותיות גדולות או קטנות. דוגמאות :

Haifa  
"Tel Aviv"  
"Kiryat Motzkin"

<limit> זהו מספר שלם חיובי אשר ממקד את היקף את הפלט המתקבל כפי שיפורט בהמשך.

תקינות הפרמטרים וברירות מחדל :  
הסדר בין הפרמטרים חייב להיות קבוע וכפי שפורט.  
האפשרויות שמותרות הן :  
כל 3 הפרמטרים מופיעים או מופיע רק הפרמטר הראשון.  
בכל מקרה אחר הפלט יהיה  
Illegal or missing parameters  
להקפיד על אותיות גדולות/קטנות, ללא שום תווים או רווחים נוספים. בין כל 2 מילים ישנו רווח בודד. בסוף שורת הפלט יורדים שורה.

אם רק קובץ המקורות (פרמטר ראשון) מופיע, אזי ברירות המחדל עבור שאר הפרמטרים הם כדלקמן :  
Haifa <city>  
1 <limit>

אם קובץ המקורות לא קיים בפועל, אזי הפלט יהיה  
File is missing  
להקפיד על אותיות גדולות/קטנות, ללא שום תווים או רווחים נוספים. בין כל 2 מילים ישנו רווח בודד. בסוף שורת הפלט יורדים שורה.

כאשר יש לנו 3 ערכים (במקור או ברירות מחדל) עבור 3 פרמטרים, אזי :  
הפלט יהיה רשימה של <limit> דירות להשכרה בעיר המבוקשת (בלבד), הזולות ביותר ביחס לכל היצע הדירות  
שניתנות בכל קבצי המקור השונים. אם יש פחות מ- <limit> דירות כאלה, אזי יוצג כמה שיש. אם יש יותר, אזי יוצגו <limit>  
הדירות הזולות ביותר.  
הדירות תוצגנה כל אחת בשורה נפרדת, כאשר השורות מסודרות ביניהן לפי תאריך עדכני, כלומר השורה הראשונה תהיה זו  
שבה מופיע התאריך העדכני (המאוחר) ביותר, והשורה האחרונה תהיה זו שבה מופיע התאריך הפחות עדכני (יותר ישן).  
אם ישנן דירות שיש להן אותו תאריך אזי הן תופענה בסדר יחסי עולה של שכ"ד, כאשר הראשונה תהיה עם שכ"ד  
היותר נמוך. דירות שיש להן אותו תאריך וגם אותו שכ"ד תופענה לפי הסדר היחסי המקורי שבו הן הופיעו במקורות, כלומר  
לפי הסדר היחסי ביניהן בקובץ המקורות ולפי הסדר היחסי הפנימי כפי שמתבטא בין הקבצים השונים ב-directories השונים.  
בכל שורה של תצוגת הפלט יוצגו הפרטים הבאים (משמאל לימין) : מספר חדרים, מחיר, טלפון.  
בין 2 ערכים סמוכים יופיע רווח בודד. אין רווחים בתחילת השורה ובסופה. בסוף שורה יורדים לתחילת שורה חדשה.  
הפלט לא מכיל שורות ריקות כלל וגם לא הערות.  
שימו לב שתיתכן כפילות בדירות המופיעות בקבצים השונים. הפלט לא יכיל כפילויות. המספר המצוין ב- <limit>  
מתייחס לדירות שונות.  
שתי שורות נחשבות ככפילות אם הן מכילות את אותן המילים בדיוק, למעט הבדלים בין אותיות גדולות/קטנות  
שמתעלמים מהם, וגם מספר הרווחים בין מילים סמוכות בשורה יכול להיות שונה עבור שורות כפולות.

## 2.3 פורמט של קובץ flat

למרות שכפי שהוזכר יש אי סנכרון אפשרי בין הסוכנויות השונות, התברר שכל הסוכנויות דאגו להקפיד על פורמט  
אחיד של קובץ flat כלשהו.  
קובץ כזה הוא קובץ טקסט. כל דירה להשכרה מופיעה בשורה נפרדת.  
תיתכנה שורות ריקות בכל מקום בקובץ. יש להתעלם מהן.  
תיתכנה גם הערות. הערה מתחילה בתו # ונמשכת ממנו ועד סוף השורה. מובטח שהתו # לא מופיע באמצע מילה,  
אבל הוא יכול להופיע צמוד לתחילת מילה שנמצאת מימין לו.  
מובטח שתוכן הקובץ תקין.

שורה שמתארת דירה להשכרה מופיעה בפורמט הבא :

< street name > <number in street> <city name> <num of rooms> <rent price> <tel number> <date of publishing>

אין ברירות מחדל לשדות השונים. ניתן להניח שהשורה תקינה ושכל השדות מופיעים. בין השדות השונים מופיעים רווחים (אחד או יותר). רווחים יכולים להופיע גם בתוך שם של רחוב או שם של עיר אשר יכולים להכיל יותר ממילה אחת.

< street name > שם של רחוב. מובטח שאינו מכיל ספרות. יכול להכיל יותר ממילה אחת.

<number in street> מספר שלם. מכיל ספרות בלבד.

<city name> שם של עיר. מובטח שאינו מכיל ספרות. יכול להכיל יותר ממילה אחת.

<num of rooms> מספר חדרים. מספר שלם. מכיל ספרות בלבד.

<rent price> שכ"ד חדשי בשקלים. מספר שלם. מכיל ספרות בלבד.

<tel number> מס טלפון.

<date of publishing> תאריך פרסום. ניתן בפורמט של d.m.y כאשר d הוא יום – ספרה או 2 ספרות, m הוא חודש – ספרה או 2 ספרות, y הוא שנה – 4 ספרות). ניתן להניח שהתאריך חוקי. לגבי d או m ייתכן 0 מוביל, אבל לא בהכרח.

דוגמות לשורות אפשריות שמתארות דירות להשכרה :

Trumpeldor 24 haifa 3 3000 052-4217542 30.3.2016

Ben Yehuda 33 Tel Aviv 4 5100 054-4288842 2.01.2016

## 2.4 דוגמות להרצה של התוכנית *getApartments*

להלן תוכנו של הקובץ sources :

#Sources of apartments for rent

e5.dat

a1.flat

b1.list.flat

d1.dat

d2.dat

ממנו מתחילים את כל התהליך. מתעלמים משורות הערה ומתייחסים רק לישויות ששמן מסתיים ב - flat. בדוגמה זו ישנן 2 ישויות כאלה, והשם שלהן מופיע ללא path בתחילתו, ולכן שתיהן מתייחסות לישויות שנמצאות ב - directory הנוכחי שבו נמצא הקובץ sources. a1.flat הוא שם של קובץ. b1.list.flat הוא שם של directory. הערה – שימו לב שאם יופיע path אזי אני לא בהכרח מתייחסים ל - directory הנוכחי.

a1.flat הוא קובץ שתוכנו הוא

#apartments in Haifa

trumpeldor 24 haifa 3 3000 052-4217542 30.3.2016

hanita 25 haifa 5 4500 053-4215365 2.1.2016

Hankin 23 Haifa 4 3200 053-7358234 6.3.2016

hankin 26 Haifa 4 4000 054-4975342 18.3.2016

trumpeldor 27 haifa 3 3100 052-4999342 24.3.2016

horev 28 Haifa 2 2950 052-4456742 4.3.2016

hagalil 29 haifa 3 3150 053-4262342 2.4.2016

HANKIN 26 haifa 4 4000 054-4975342 18.3.2016

trumpeldor 27 HAIFA 3 3100 052-4999342 24.3.2016

horev 28 haifa 2 2900 052-4456742 4.3.2016

# an old apartment - hankin 56 Haifa 4 4000 054-4975342 18.3.2010

b1.list.flat הוא directory אשר יש בו קובץ b2.flat אשר תוכנו הוא

# This includes also apartments in Tel Aviv

arlozorov 31 Tel Aviv 5 6650 053-4212142 2.1.2016

even gvirol 32 tel aviv 6 6800 052-4215542 30.3.2016

ben yehuda 33 Tel Aviv 4 5100 054-4288842 30.3.2016

yfe nof 34 haifa 3 3500 052-4218342 30.4.2016

hankin 26 haifa 4 4000 054-4975342 18.3.2016

trumpeldor 27 haifa 3 3100 052-4999342 24.3.2016

horev 28 Haifa 2 2900 052-4456742 4.3.2016

horev 29 Haifa 2 2950 052-4456896 4.3.2016

hayarkon 10 Tel Aviv 1 1800 053-7358234 6.3.2016

**דוגמה 1 :**

getApartments

הפלט המתקבל :

Illegal or missing parameters

הסבר : חובה לתת פרמטר אחד או שלושה פרמטרים.

## דוגמה 2 :

getApartments data "TEL AVIV" 50

הפלט המתקבל :

File is missing

הסבר : הקובץ data לא קיים בדוגמה זו, ולכן מקבלים הודעת שגיאה.

## דוגמה 3 :

getApartments sources "TEL AVIV" 3

הפלט המתקבל :

4 5100 054-4288842  
1 1800 053-7358234  
5 6650 053-4212142

הסבר : סה"כ בת"א יש 4 דירות בהיצע הכולל, אבל כאן מבקשים רק 3, ולכן מוצגות 3 הדירות הזולות ביותר, אבל לפי סדר יורד של תאריכים. הדירה של 5100 – התאריך שלה הוא 30.3, הדירה של 1800 – התאריך שלה הוא 6.3, הדירה של 6650 – התאריך שלה הוא 2.1.

## דוגמה 4 :

getApartments sources HAIFA 1

הפלט המתקבל :

2 2900 052-4456742

הסבר : מוצגת הדירה הזולה ביותר בחיפה. שימו לב לה שהדירה הזו מופיעה גם ב-a1.flat וגם ב-b2.flat. אבל הפלט לא מכיל כפילויות, ולכן בפלט הדירה מופיעה פעם יחידה. כפילות נבדקת תוך התעלמות מהבדלים בין אותיות גדולות/קטנות ותוך התעלמות ממספר שונה של תווי רווח בין מילים סמוכות.

## דוגמה 5 :

getApartments sources HAIFA 4

הפלט המתקבל :

3 3000 052-4217542  
2 2900 052-4456742  
2 2950 052-4456742  
2 2950 052-4456896

הסבר : שימו לב של-3 הדירות התחתונות מתוך ה-4 יש אותו תאריך – 4.3.2016. אזי קודם כל מופיעה הדירה הזולה (2900) ואחר כך יש 2 דירות שלהן אותו תאריך וגם אותו מחיר, ולכן הן מופיעות לפי הסדר היחסי כפי שהופיעו בקבצי המקורות. קודם מופיעה הדירה שהופיעה ב-a1.flat ואחר כך מופיעה הדירה שהופיעה ב-b2.flat.

הערה : כל הדוגמות הנ"ל ניתנות ב - [~mtm/public/1516b/ex3/](https://github.com/~mtm/public/1516b/ex3/) .  
הן גם משמשות כבדיקות לדוגמה ב - `final_check` ( ראו סעיף 4 בהמשך במסמך זה ).

## 2.5 הנחיות להכנת התרגיל

התוכנית שהנכם נדרשים לכתוב במסגרת תרגיל זה הינה מודולרית וצריכה לכלול מספר תוכניות `script` אשר קוראות אחת לשניה. כל תוכנית כזו צריכה גם היא להיות מודולרית ולכלול, בין היתר, גם פונקציה או פונקציות.

התוכנית צריכה להיכתב כך **שאין צורך כלל** להשתמש בקבצים זמניים שיוצרים במהלך ריצת התוכנית. השתמשו במנגנון ה - `Pipe`.

**חובה** לצרף להגשה קובץ `readme.txt` שבו נדרש לכתוב בקצרה (טקסט אנגלית בלבד) איזה קבצי `script` קיימים בתוכנית שכתבתם. לכל קובץ `script` נדרש לציין איזה פרמטרים הוא מקבל ומה משמעותם ובנוסף נדרש לציין מה הוא מבצע. בנוסף נדרש לציין עבור כל קובץ `script` איזה פונקציות מוגדרות בו, ובנוסף, לכל פונקציה שמוגדרת בו נדרש לפרט איזה פרמטרים היא מקבלת (אם בכלל) ומה משמעותם ובנוסף - מה הפונקציה עושה.

התוכנית שתגישו חייבת לכלול, בכל מקרה, את 3 קבצי ה-`script` הבאים (חייבים להקפיד על שמות מדויקים כפי שמוגדרים כאן, כולל אותיות גדולות/קטנות) :

`cleanLines` - זהו `script` עזר אשר קורא מהקלט הסטנדרטי שורות טקסט והוא מסנן מתוכן שורות ריקות וגם את כל ההערות, כפי שתואר בסעיפים הקודמים.

`handleNamesDates` - זהו `script` עזר אשר מופעל בשלב מאוחר יותר לאחר `cleanLines`. הוא קורא שורות טקסט שמועברות לו באמצעות הקלט הסטנדרטי והוא מסדר את תוכן השורות כך שיהיה קל להתמודד עם שמות רחובות/ערים מרובי מילים ועם תאריכים אשר מופיעים בשורות השונות (למשל - מטפל ברווחים שאולי מפריעים לצרכים שלנו וכו').

במסגרת הגדרת התרגיל לא ניתן כאן פירוט מדויק של מה בדיוק עושים כאן. הסטודנטים יכולים להגדיר כאן מה שמתאים להם. במקרה הקיצוני אפשר להגדיר כאן `script` שלא עושה דבר אלא מעביר את הקלט לפלט כמו שהוא. כפי שכבר צוין, הנכם מתבקשים לתאר בקובץ `readme.txt` את המבנה המודולרי של התוכנית שכתבתם.

`getApartments` : זוהי תוכנית ה-`script` הראשית. היא מקבלת 3 פרמטרים כפי שמתואר בסעיף 2.2.

תוכנית זו בודקת את תקינות הפרמטרים שהיא מקבלת בהתאם למה שמתואר בסעיף 2.2.

לאחר מכן היא קוראת את השורות הנדרשות מהמקורות השונים בהתאם לפרמטרים שהיא מקבלת

ומשתמשת בתוכניות עזר ובפקודות של Unix כדי לבצע מה שצריך.

השתמשו בפונקציית עזר שתגדירו לצורך קריאת השורות הנדרשות מהמקורות השונים.

השתמשו במנגנון ה-`Pipe` כדי לבנות באופן מודולרי את תהליך העבודה הרצוי עד לקבלת התוצאות הסופיות.

כפי שיפורט בסעיף 5, בתוכנית שתכתבו ניתן להניח שמשתנה הסביבה `PATH` מכיל את ה-`directory` הנוכחי שבו נמצאים. ניתן להניח שכל קבצי `script` שאתם מגישים במסגרת התרגיל נמצאים במיקום הנוכחי. אין צורך להוסיף /. כאשר קוראים להם.

הערה - בבקשה לא להתבלבל כאן. קבצי המקור שמהם קוראים את נתוני הדירות לא מובטחים להיות ב-`directory` הנוכחי.

### 3 הגשה

את ההגשה יש לבצע דרך אתר הקורס, תחת:

Assignments → HW3 - BASH Scripting → Electronic Submit

עליכם להגיש קובץ zip אשר מכיל בתוכו את כל קבצי ה-script ובנוסף הוא מכיל קובץ readme.txt כפי שפורט. בנוסף חובה לצרף בתוך קובץ zip את הקובץ student.txt ממולא באנגלית בלבד. הקובץ פורסם ביחד עם התרגיל. הקפידו למלא באופן מדויק כתובת דוא"ל. לכתובת זו ישלח דוא"ל שיכיל את פירוט התוצאות של הבדיקה האוטומטית שתבוצע.

הערה – final\_check כפי שיפורט בהמשך יבדוק את קיומם של קבצים אלה בקובץ ה-zip. נדרש להקפיד על קובץ zip ולא קובץ rar. אסור שבקובץ zip תופיע תיקייה כלשהי.

הקפידו על שמות מדויקים של הקבצים כפי שמפורט בסעיף 2.5. בין היתר הקפידו על אותיות גדולות/קטנות. זכרו ששמות הקבצים הם case sensitive במערכות הפעלה דמויות Unix, ושטעות קטנה בשם של קובץ script עלולה להכשיל את הבדיקה.

**חובה לשמור את קוד האישור אשר מתקבל בעת ההגשה, וחובה לשלוח אותו לשותף (אם ישנו שותף).** אם לא תעשו זאת - אתם חשופים למצב שבו אתם עלולים לקבל 0 בתרגיל על שטות (למשל - 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 Error! Reference source not found.**

### 4 בדיקה עצמית לפני הגשה - final\_check

על מנת לוודא שקובץ ההגשה שלכם תקין. לפני שאתם מגישים, עליכם להריץ דרך חשבון ה-T2 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-T2):

```
>~mtm/public/1516b/ex3/final_check <submission file>
```

כאשר <submission file> הוא קובץ ה-zip שאתם מעוניינים להגיש.

לדוגמה, עבור קובץ הגשה בשם hw3.zip, כך נראית הרצה מוצלחת של final\_check:

```
>~mtm/public/1516b/ex3/final_check hw3.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

Everything went well :)

Final check passed.



## 5 דרישות, הגבלות והערות כלליות

- שימו לב לנושא "ייצוג ה-newline" במסמך "[מידע כללי לגבי תרגילי הבית](#)" (סעיף 1.3), ואל תשכחו להשתמש בתוכנית dos2unix כאשר אתם מעבירים קבצי טקסט מ-Windows לסביבת Unix.
- **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](#)" (סעיף 1.2).
- **אסור להשתמש בתוכניות Unix סטנדרטיות אשר לא נלמדו בתרגולים, וגם לא בדגלים שלא נלמדו.**
- היעזרו בתוכניות הסטנדרטיות שנלמדו (cut, uniq וכו'). שימוש בפקודות אלה יכול לקצר בהרבה את כתיבת הסקריפט!
- הסקריפטים ייבדקו על T2.
- העובדה שהבדיקה תבצע על T2 אינה אומרת שאתם חייבים כל הזמן לפתח על T2. אתם יכולים לפתח את הסקריפטים שלכם במחשבים אחרים ובהמשך להעביר אותם ל-T2 ולוודא שהם עובדים נכון על ה-T2. **שימו לב** – סקריפטים אשר עובדים נכון במחשב אחר עלולים ליצור פלט שגוי על T2 – לפני ההגשה יש לוודא שהסקריפטים שלכם עובדים נכון על ה-T2! (ובפרט יש להריץ את סקריפט הבדיקה הסופית)
- 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 T2 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 T2. צוות הקורס לא יוכל לטפל בבעיות שקשורות לעומס/איטיות ב-T2. לא יתנו שום הקלות/דחיות בגלל העומס/איטיות של T2.**
- 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 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 ל-stdout/stderr, **טעות קלה של תו רווח יחיד תכשיל בדיקה שלמה.**
- בזמן תכנון הסקריפטים שלכם - השתדלו כמה שיותר להימנע מכפילויות בקוד (מופעים שונים של אותו קוד בשינויים קלים). עיקרון זה הינו בעל חשיבות אדירה במת"ם ובכלל בחיים, והוא לא רק יחסוך לכם 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.
- 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/Spring2016/en/news.html>  
מומלץ להתעדכן בעמוד ה-F.A.Q של התרגיל (בדף זה יפורסמו שאלות ותשובות נפוצות בנוגע לתרגיל).

**בהצלחה !**