

מבוא למדעי המחשב - תרגיל 3

[מבוא](#)

[הנחיות לפיתרון התרגיל](#)

[משימות](#)

[מציאת מספר מקסימלי](#)

[מציאת המספר השני הקטן ביותר](#)

[סכימת קלטים עד לתנאי עצירה](#)

[חישוב מסלול דו מימדי](#)

[חישוב אברי סידרה](#)

[פירוק מספר לספרותיו](#)

[המרת מספר בינארי לעשרוני](#)

[המרת מספר עשרוני לבינארי](#)

[המרה כללית בין בסיסי ספירה](#)

[נהלי הגשה](#)

[בדיקת התרגיל טרם הגשתו](#)

מבוא

בכל סעיף להלן מתוארת בעייה.

- את הפיתרון לכל סעיף ממשו בקובץ נפרד ותנו לו שם כמופיע בסעיף.
- בסעיפים הדורשים הדפסות למסך, עשו שימוש מדויק במחרוזות המסופקות.
 - כל הוראה להדפסה למסך מתייחסת להדפסה בשורה חדשה.
- תאריך ההגשה לתרגיל: 20.11.2014, שעת ההגשה 21:00.

הנחיות לפיתרון התרגיל

בכל הסעיפים יש להקפיד במדויק על ההוראות הנתונות בתרגיל, בפרט:

- על שמות הקבצים להיות בדיוק אלו המצויינים, בפרט:
 - על הקבצים להיות עם סיומת py לכל קבצי הפייטון וללא סיומת לקובץ ה- README.
 - שימו לב לאותיות גדולות וגדולות.
- ההדפסות למסך צריכות להיות במדויק כמצויין בתרגיל, בפרט:
 - הקפידו לא להוסיף ולא להחסיר רווחים.
 - סימני הפיסוק צריכים להיות זהים לאלו המופיעים בהנחייה.
- סגנון:
 - הקפידו על תיעוד נאות.
 - בחרו שמות משתנים משמעותיים.
 - זכרו לא לעשות שימוש ב"מספרי קסם". מספרים אשר עושים בהם שימוש במהלך התרגיל צריכים להיות מוגדרים כמשתנים.
- Headers - אפשר, אך אין חובה לכלול header בתחילת קבצי מימוש.
- פיתרון משימות 7-9 : טרם פיתרון כל אחת מהמשימות, מומלץ לקרוא קודם את שלושתן ורק אז לגשת לפתרון.

- התרגיל תוכנן כך שניתן לפתור אותו במלואו ללא שימוש ברשימות, אך אין מניעה לעשות בהן שימוש.

משימות

1. מציאת מספר מקסימאלי

פרודו וסם יושבים על ראש גבעה ומצפים לבואו של גנדלף. לפתע מופיעה לפנייהם שיירה של רוכבים וסם אומר לפרודו כי גנדלף הוא הרוכב בעל המצנפת הגבוהה ביותר ומפרט בקול את גובה המגבעות של כל העוברים בשיירה לפי הסדר. עזרו לפרודו לזהות באיזה מקום בשיירה נמצא גנדלף.

- שם קובץ המימוש - findLargest.py
- הדפסו הודעה למשתמש לקבלת מספר הרוכבים בשיירה :
- Enter the number of riders:
- עבור כל אחד מהרוכבים הדפסו למסך הודעה לקבלת גובה המצנפת:
- How tall is the hat?
- בסיום הקלט, הדפסו למסך הודעה המכריזה על מיקומו של גנדלף בשיירה:
- Gandalf's position is: X
- כאשר X מוחלף במספרו של גנדלף בשיירה.
- לדוגמה אם גנדלף הוא הרוכב הראשון ברשימה יש להדפיס:
- Gandalf's position is: 1
- הנחות על הקלט :
- ניתן להניח כי הקלט על מספר הרוכבים הוא מספר שלם חיובי.
- ניתן להניח כי הקלטים על גבהי המצנפות הם מספרים חיוביים, שונים כולם.
- הגבלות על המימוש - את מציאת המספר הגדול ביותר עליכם לממש בעצמכם ללא שימוש בפונקציות מובנות של Python.

2. מציאת המספר השני הקטן ביותר

גנדלף הבטיח לתת לפיפין זיקוקים, אבל לא בטוח מי הוא פיפין. סם אומר לגנדלף כי פיפין הוא השני הכי צעיר מבין עשרת הרוקדים בשורה על הבמה. כלומר קיים רוקד אחד בדיוק אשר הוא צעיר יותר מפיפין.

- שם קובץ המימוש - findSecondSmallest.py
- עבור כל אחד מעשרת הרוקדים הדפסו למסך הודעה לקבלת גילו:
- What is the age of the current dancer?
- בסיום הקלט, הדפסו למסך הודעה המכריזה על מיקומו של פיפין:
- Pippin is dancer number X
- כאשר X מוחלף במספרו של פיפין בשורה.
- לדוגמה אם גילו של פיפין הוכנס כקלט הראשון יש להדפיס:
- Pippin is dancer number 1
- הנחות על הקלט :
- ניתן להניח כי מוכנסים עשרה קלטים וכי הם כולם מספרים חיוביים ושונים.
- הגבלות על המימוש - את מציאת המספר השני הקטן ביותר עליכם לממש בעצמכם ללא שימוש בפונקציות מובנות של Python.

3. סכימת קלטים עד לתנאי עצירה

סם ופרודו אורזים תיק ליציאה למסע. סם מכניס פריטים לתיק ואומר את משקלם בקול. הפריט האחרון שסם מכניס לתיק הוא טבעת שמשקלה 1- (מינוס אחד). גנדלף לא מרשה להם לקחת תיק שסך משקל הפריטים הארוזים בו גדול מ- 100.

- שם קובץ המימוש - totalWeight.py
- הדפסו למסך הודעה יחידה:

Insert weights one by one:

- לאחר ההדפסה יש לקלוט קלט מהמשתמש (ללא הדפסה עבור כל פריט) המייצג את משקל הפריט הנוכחי.
- אם סך המשקל בתיק עובר את המותר הדפסו למסך הודעה וסיימו את ריצת התוכנית. ההודעה להדפסה:

Overweight! Gandalf will not approve.

- אם לא הסתיימה הריצה בשל משקל עודף והגיע הפריט האחרון (הטבעת), הדפסו למסך לאחר סיום קבלת הפריטים הודעה המכריזה על סך משקל הפריטים בתיק, ללא חישוב הטבעת:

The total packed weight is X

- כאשר X מוחלף במשקל סך הפריטים ללא חישוב הטבעת.
- לדוגמה אם הוכנס פריט אחד שמשקלו 8 ומיד אחריו הוכנסה הטבעת, המשקל הכולל הוא 8 וההודעה להדפסה היא:

The total packed weight is 8

- משקלם של פריטים חייב להיות אי שלילי.

- אם הכניס המשתמש משקל שלילי עבור פריט (שאינו הטבעת) יש להדפיס למסך את הודעה:

Weights must be non-negative

- לאחר מכן יש להמשיך בקבלת קלטים ללא סכימת האיבר (הנוכחי) בעל המשקל השלילי.
- לדוגמה אם הקלטים המוכנסים הם (משמאל לימין, אחד אחרי השני): [8,-2,3,-1], הפלט המבוקש הוא (שורות המתחילות בתו "#") לא צריכות להיות מודפסות למסך ומוספות רק לצורך ההסבר):

Insert weights one by one:

#User inputs 8

#User inputs -2

Weights must be non-negative

#User inputs 3

#User inputs -1

The total packed weight is 11

- הנחות על הקלט :

- ניתן להניח כי המשקל המוכנס עבור כל הפריטים הוא מספר.
- משקל כל הפריטים מלבד הטבעת שונה מ- 1- (מינוס אחד).

4. חישוב מסלול דו מימדי

סם ופרודו יוצאים באישון לילה למסע ורוצים להשאיר הוראות לגנדלף איך להגיע אליהם. בכל פנייה בדרך הם כותבים לאן פונה הדרך (ימינה או שמאלה) ואת מספר הצעדים אותם צעדו עד לפנייה הבאה. כאשר הם מגיעים ליעדם הם מציינים את הפנייה האחרונה כסוף. כאשר גנדלף מגיע לפגוש את סם ופרודו הוא עף על גבי נשר ולכן מספיק לו לדעת את מספר הצעדים הסופי קדימה/אחורה וימינה/שמאלה כדי להגיע אליהם.

○ שם קובץ המימוש - twoDimensionalSeek.py

○ כל עוד לא הגיעו סם ופרודו ליעדם הדפיסו למסך בקשה לקבלת כיוון הפנייה הנוכחי:

Next turn:

■ אם סם ופרודו פונים ימינה הם מכניסים את הקלט "right" ואם שמאלה את הקלט "left".

■ הקלט אשר מסמל את הגעתם ליעד הוא המחרוזת "end" כקלט לכיוון הפנייה.

○ לאחר קבלת כיוון הפנייה הנוכחית (אם עדיין לא הסתיים המסע) יש להדפיס הודעה למסך לקבלת מספר הצעדים שנצעדו לאחר הפנייה.

How many steps?

○ לאחר סיום קבלת כל הקלטים יש להדפיס הודעה לגנדלף על המסלול אותו הוא צריך לעוף:

Gandalf should fly X steps [right\left] and Y steps [forward\backward]

○ כאשר X מוחלף במספר הצעדים אותו יש לעוף או ימינה או שמאלה ו - Y במספר הצעדים אותו יש לעוף קדימה או אחורה.

■ נקודת ההתחלה למסע היא כאשר סם ופרודו עומדים בפני פיצול דרכים ראשון בו הם בוחרים ללכת ימינה או שמאלה.

■ אם מספר הצעדים הסופי לימין ושמאל או לקדימה ואחורה הוא 0 יש להצהיר 0 צעדים ימינה וקדימה בהתאמה.

■ לדוגמה אם פרודו וסם פנו ימינה והלכו 10 צעדים ואז שמאלה 20 צעדים ההדפסות למסך הן (תווים בצבע כחול מסמלים את הקלט שמכניס המשתמש):

Next turn:right

How many steps?10

Next turn:left

How many steps?20

Next turn:end

Gandalf should fly 10 steps right and 20 steps forward

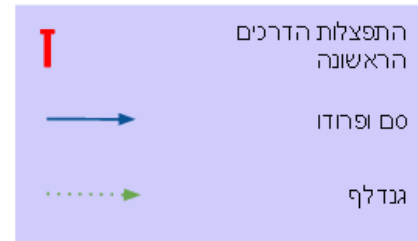
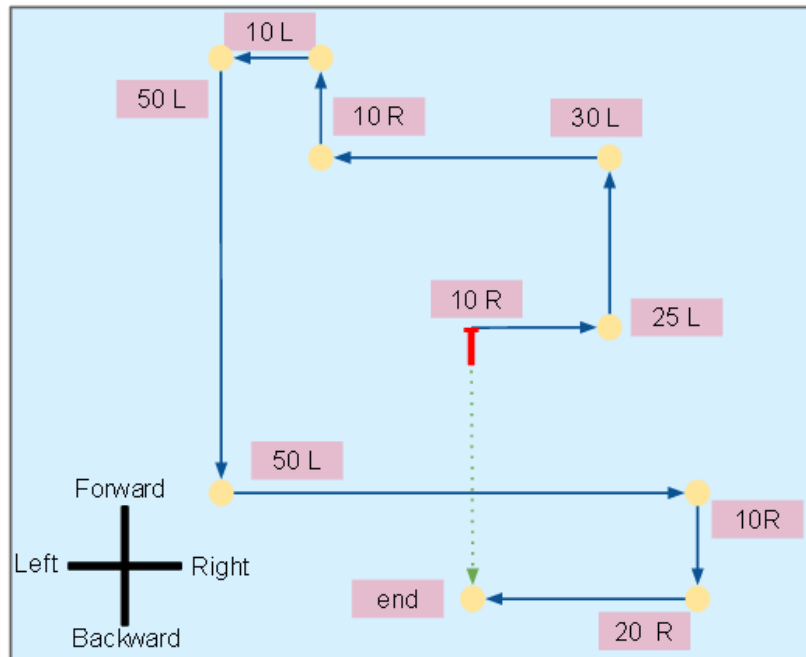
■ דוגמה נוספת, אם פרודו וסם פנו ימינה והלכו 10 צעדים, שמאלה 10 צעדים, ימינה 10 צעדים וימינה עשרה צעדים ההודעה הסופית להדפסה היא:

Gandalf should fly 20 steps right and 0 steps forward

○ הנחות על הקלט :

■ ניתן להניח כי בעת הכנסת הקלט לכיוון, הקלט הוא תמיד אחת משלוש המחרוזות "left", "right", או "end".

■ ניתן להניח כי בעת הכנסת הקלט למספר הצעדים הקלט הוא תמיד מספר שלם אי שלילי.



Gandalf should fly 0 steps right and 25 steps backward

5. חישוב אברי סידרה

לגולס מכין חצים כדי לשלח אותם באורקים. האורקים נלחמים בלגולס אחד אחרי השני ולומדים מהטעויות של קודמיהם כך שמספר החצים להם נזקק לגולס על מנת להכריע אורק מסויים שווה לסכום מספר החצים לו נזקק כדי להכריע את שני האורקים הקודמים. את שני האורקים הראשונים מכריע לגולס בחץ בודד, אך הוא מעוניין לברר לכמה חצים יזדקק עבור אורק מסויים אחר ברשימה.

- שם קובץ המימוש - `ithElementValue.py`
- הדפיסו למסך הודעה יחידה:

Which Orc do you wish to confront?

- הדפיסו למסך הודעה יחידה על מספר החצים בהם יש צורך כדי להכריע את מספר האורק שהתקבל. ההודעה להדפסה:

The required number of arrows is X

- כאשר X מוחלף במספר החצים הדרוש.
- לדוגמה אם מספר האורק המוכנס הוא 5, אזי עבור האורק השלישי היה צורך ב - 2 חצים (סכום החצים עבור שני האורקים הראשונים) עבור האורק הרביעי ב - 3 חצים ועבור האורק החמישי ב - 5 חצים לכן יש להדפיס:

The required number of arrows is 5

- הנחות על הקלט :
- ניתן להניח כי הקלט למספר האורק הנו מספר שלם חיובי.

6. פירוק מספר לספרותיו

גימלי מבצע מעקב על מספר כוסות השיכר שהוא שותה מדי יום על ידי כתיבת מספר הכוסות אותן שתה כל יום מימין לשמאל. כיוון שהוא מרבה לשתות, גימלי שכח להכניס פסיקים בין מספרי כל הימים ולכן לדוגמה במקום לציין שהוא שתה 7 כוסות ביום הראשון, 5 ביום השני ו - 9 ביום השלישי, יש בידו את המספר היחיד 957.

- שם קובץ המימוש - decomposition.py
- הדפסו למסך הודעה יחידה:

Insert composed number:

- עבור כל אחת מהספרות במספר יש להדפיס הודעה המצהירה על כמות כוסות השיכר אותן שתה גימלי ביום זה:

The number of goblets Gimli drank on day X was Y

- כאשר X מוחלף במספר היום בו נשתו הכוסות - כאשר הספרה הראשונה מימין (לדוגמה - 7 במספר - 957) מציינת את היום הראשון וכן הלאה, ו - Y מציין את מספר הכוסות שנשתו ביום זה.

- לדוגמה אם המספר המוכנס הוא 957, אזי ההודעות אותן יש להדפיס הן:

The number of goblets Gimli drank on day 1 was 7

The number of goblets Gimli drank on day 2 was 5

The number of goblets Gimli drank on day 3 was 9

- הנחות על הקלט :

- ניתן להניח כי הקלט הנו מספר חיובי שלם.
- ניתן להניח כי גימלי שותה מדי יום מספר חד ספרתי של כוסות שיכר.
- אפשר להניח כי הספרות השמאליות ביותר (בפרט ספרה אחת) שונות מ - 0 (אפס). לדוגמה אין צורך לטפל באופן מיוחד במספרים 025 או 007 ואפשר להתייחס אליהם כאל המספרים 25 ו - 7 בהתאמה.
- בהתאמה, אל המספר שכל ספרותיו הן 0 (0, 00, 000 ...) יש להתייחס כמספר 0.

- הנחיות למימוש : על המימוש להתייחס למספר המוכנס כ**מספר** ולא כרצף. בפרט, בתשובות אין להמיר את המספר למחרוזת או טיפוס אחר ולבדוק מהו התו (character) במקום כלשהו.

7. המרת מספר בינארי לעשרוני

בני הלילית רוצים להציע עזרה על ידי תרומה של ספינות והם מעבירים לאראגורן מסמך עם מספר הספינות אותו הם יכולים לספק. לרוע המזל, בני הלילית משתמשים בכתובי בינארי ואראגורן לא מבין כמה ספינות הם מוכנים לתת.

- שם קובץ המימוש - binaryToDecimal.py
- הדפסו למסך הודעה יחידה:

Insert number in binary representation:

- הדפסו למסך הודעה יחידה עם ערכו העשרוני של המספר:

The decimal value of the inserted binary number is X

- כאשר X מוחלף בערכו העשרוני של המספר המוכנס.
- לדוגמה אם המספר המוכנס הוא 110, אזי ערכו העשרוני הוא 6 לכן הודעת ההדפסה צריכה להיות:

The decimal value of the inserted binary number is 6

- הנחות על הקלט :

- ניתן להניח כי הקלט הנו מספר חיובי שלם המכיל את הספרות '0' ו - '1' בלבד.
- הגבלות על המימוש - את ההמרה מכתובי בינארי לעשרוני עליכם לממש בעצמכם ללא שימוש בפונקציות מובנות של Python.

8. המרת מספר עשרוני לבינארי

אראגורן נרגש כל כך מפנייתם של בני הלילית עד כדי כך שהוא כותב להם מכתב בו הוא מבטיח כי ירשום להם שיר עם מילים רבות. לרוע המזל בני הלילית לא יבינו כמה מילים ארגורן מתכוון לכתוב בשיר כי הם מבינים רק כתיב בינארי.

- שם קובץ המימוש - decimalToBinary.py
- הדפיסו למסך הודעה יחידה:

Insert number in decimal representation:

- הדפיסו למסך הודעה יחידה עם ערכו הבינארי של המספר:

The binary value of the inserted decimal number is X

- כאשר X מוחלף בערכו הבינארי של המספר המוכנס.
- לדוגמה אם המספר המוכנס הוא 6, אזי ערכו הבינארי הוא 110 לכן הודעת ההדפסה צריכה להיות:

The binary value of the inserted decimal number is 110

- הנחות על הקלט :
- ניתן להניח כי הקלט הנו מספר חיובי שלם בייצוג עשרוני.
- הגבלות על המימוש - את ההמרה מכתוב עשרוני לבינארי עליכם לממש בעצמכם ללא שימוש בפונקציות מובנות של Python.

9. המרה כללית בין בסיסי ספירה

שאלה זו הינה שאלה תאורטית ואין צורך בקוד על מנת לפתור אותה. את התשובה לשאלה זו יש לכתוב באנגלית בקובץ ה - README.

- תארו במילים מהו השוני בין המימוש למשימה 7 (המרה ממספר בינארי לעשרוני) ומשימה 8 (המרה ממספר עשרוני לבינארי). האם קיים שוני מהותי בין שני המימושים? אם כן, מה הוא?
- מה השינוי המינימאלי שאפשר לעשות במימוש משימה 7 על מנת שתפתור את משימה 8? ולהפך?
- אילו היינו רוצים לכתוב תכנה הממירה מספרים מבסיס 6 לבסיס 2 או מבסיס 4 לבסיס 10, האם היינו יכולים, בשינויים קלים, להשתמש בקוד הפותר את משימה 7 או משימה 8? האם ניתן לכתוב תכנית כללית המקבלת מהמשתמש מספר בבסיס כלשהו קטן מעשר $base_source$ וממירה אותו לבסיס אחר קטן מעשר $base_target$ (מבלי לכתוב קוד ספציפי הממיר בין כל הזוגות האפשריים)?

נהלי הגשה

- יש להגיש קובץ יחיד בשם ex3.tar דרך [הקישור המתאים](#) באתר הקורס. על קובץ זה לכלול את כל הקבצים הבאים בלבד:

1. README (כמוסבר באתר הקורס על פי הדוגמה הנתונה שם).
2. findLargest.py
3. findSecondLargest.py
4. totalWeight.py
5. twoDimensionalSeek.py
6. ithElementValue.py
7. decomposition.py

binaryToDecimal.py .8

decimalToBinary.py .9

- על מנת ליצור את קובץ ההגשה ex3.tar ניתן להשתמש בפקודה הבאה :

```
tar cvf ex3.tar README findLargest.py findSecondSmallest.py
totalWeight.py twoDimensionalSeek.py ithElementValue.py
decomposition.py binaryToDecimal.py decimalToBinary.py
```

- כדי לוודא שקובץ הארכיב יוצר בהצלחה ניתן לחלץ את תוכנו לספריה מקומית ולוודא את תוכנו בעזרת הפקודה :

```
tar xvf ex3.tar
```

בדיקת התרגיל טרם הגשתו

התיקיה ~/intro2cs/bin/ex3 מכילה פתרונות מקומפלים לכל משימות, בעזרתם תוכלו לבדוק כי המימוש שלכם מניב אותם פלטים כמצופה.

- **פתרונות מקומפלים :** בתיקיית הפתרונות נמצא קובץ פתרון ללא סיומת עבור כל אחת מהמשימות (לדוגמה findLargest, findSecondSmallest וכו').

- לא ניתן לראות את הקוד שבקבצים אלה אך ניתן להריצם כרגיל על ידי הקלדת שמם. כך לדוגמה על מנת להריץ את הקובץ findLargest יש להריץ את הפקודה:

```
<1|0>usr:~/intro2cs/bin/ex3/findLargest
```

- **האופרטור "<"** בלינוקס מכניס קובץ תוכן מקור המצוין בצד ימין לצד שמאל.

- כך לדוגמה אם קיים בספרייה הנוכחית קובץ בשם input.txt שתוכנו הוא (משמאל לימין, בכל שורה ספרה אחת) : 3,1,2,3. הרצת הפקודה הבאה :

```
<2|0>usr:~/intro2cs/bin/ex3/findLargest < input.txt
```

- תריץ את הקובץ findLargest ותכניס לו כפלט את הערכים המצויים בשורות לפי הסדר (הערך הראשון המוכנס הוא 3 לאחר מכן 1 ואז 2 ואז 3).

- **האופרטור ">"** בלינוקס מכניס תוכן מקור המצוין מצד שמאל שלו ליעד המצוין מצד ימין שלו. בפרט, ציון תוכן מקור כתוכנית python ויעד כקובץ טקסט, יכניס את כל הפלט שהייתה מייצרת תכנת ה - python למסך אל תוך קובץ הטקסט.

- כך לדוגמה הרצת הפקודה הבאה בלינוקס תריץ את הקוד המקומפל הנמצא בקובץ findLargest, תכניס לו כפלט (שורה אחר שורה) את תוכן הקובץ input.txt וכל פלט שתכנה זו תוציא לא יודפס למסך כי אם ייכנס לקובץ output.txt :

```
<3|0>usr:~/intro2cs/bin/ex3/findLargest < input.txt > output.txt
```

- **הפקודה diff** משווה בין שני קבצי טקסט ומזהה אי התאמות ביניהם (להסבר נוסף ראו [כאן](#)).

- כך לדוגמה עבור שני הקבצים mySolution.txt ו - schoolSolution.txt הפקודה :

```
diff mySolution.txt schoolSolution.txt
```

- תציין את ההבדלים בין שני הקבצים אם קיימים כאלה. ■ אם שני הקבצים זהים, לא יודפס למסך כלום :

```
<4|0>usr:~/intro2cs/bin/ex3%diff mySolution.txt schoolSolution.txt
```

```
<5|0>usr:~/intro2cs/bin/ex3%
```

- הסבר תמציתי על איך לקרוא את הפלט של diff אפשר למצוא [כאן](#).

- אפשרות חלופית לשימוש ב - diff, אשר עושה שימוש בממשק גרפי בו מוצגים ההבדלים בין הקבצים בצבע, היא הפקודה [vimdiff](#).

- על מנת לבדוק אם הפיתרון שלכם מניב אותן תוצאות בדיוק כפי שמייצרים הפתרונות המקופלים, ניתן לכתוב קובץ קלט, להריץ אותו עבור הפיתרון שלכם והפיתרון המקופל, לייצא את הפלטים לקובץ ולבדוק בעזרת diff כי הפלטים זהים.

○ כך לדוגמא אם נרצה לבדוק את המימוש שלנו של findLargest.py לפיתרון המקומפל המפורסם findLargest:

- ניצור קובץ input.txt אשר יכיל טקסט המתאים כקלט לבעייה זו. לדוגמה משמאל לימין, בכל שורה ספרה אחת : 3,1,2,3.
- נריץ את התכנה שלנו עם קובץ הקלט ונייצא אותו לקובץ פלט (דוגמה myOutput).
- נריץ את הפיתרון המקומפל על אותו הקלט ונייצא אותו לקובץ פלט אחר (דוגמה schoolOutput).
- נריץ diff על שני קבצי הפלט, אם הפלט של פקודת ה - diff הוא ריק, אזי סימן הוא שהקבצים זהים.

```
python3 findLargest.py < input.txt > myOutput
~intro2cs/bin/ex3/findLargest < input.txt > schoolOutput
diff myOutput schoolOutput
```

% Hopefully nothing was printed to screen :-)

בהצלחה!