

מערכות תוכנה

מטלה 3

מערכים ומחרוזות

אחראי מטלה: יבגני הרשקוביץ נייטרמן
Neiterman21@gmail.com

שימו לב:

- המטלה היא בזוגות או יחידים – לא שלשות.
- עליכם לבצע את פקודת הקומפילציה עם הדגל – Wall על מנת לוודא שתוכניתכם מתקמפלת ללא אזהרות. תכנית שמתקמפלת עם אזהרות תגרור הורדת נקודות.
- עליכם לוודא שתוכניתכם מתקמפלת ורצה על גבי מערכת ההפעלה ubuntu עם קומפיילר gcc
- יש להגיש את המטלה ב-git . יש להגיש קובץ txt. השורה הראשונה תכלול את הכתובת של ה-git (לא URL) מאתר ה-github. השורה השניה תכלול את מזהה ה-commit הרלוונטי והשורה השלישית את תעודות הזהות של הסטודנטים המגישים מופרדים ברווח.
- הנכם נדרשים לקוד קריא ונקי.
- בכל מקום בו יש צורך בשימוש בקבועים בעלי משמעות יש להגדיר אותם באמצעות define

שאלה 1. מיון הכנסה 30 נקודות

בשאלה זו נממש אלגוריתם של מיון הכנסה כמו שראיתם באלגוריתמי מיון. למי שלא זוכר או לא למד ניתן לקרא בויקיפדיה [מיון הכנסה](#).

בשאלה זו בלבד חל איסור להשתמש בכתיב מערכים (כלומר גישה לאיבר עם סוגריים מרובעים []). ניתן להשתמש ב[] רק בשורת ההצרה על המערך כשקובעים את גודלו. לאחר מכן כל גישה למערך חייבת להיות עם אריתמטיקה של מצביעים: $*(arr + i)$

סעיף א shift_element 10 נקודות.
יש לממש את הפונקציה עם החתימה:

```
shift_element(int* arr, int i)
```

הפונקציה תקבל מצביע למערך (לא בהכרח לתחילת המערך) ומספר הפונקציה תזיז את i האיברים הבאים במערך תא אחד ימינה. שימו לב כי הפונקציה מניחה כי קיים זיכרון ללפחות i+1 תאים אחרי המצביע למערך.

סעיף ב insertion sort 20 נקודות
עליכם לממש את הפונקציה עם החתימה:

```
insertion_sort(int* arr, int len)
```

על הפונקציה לקבל מערך בגודל len ולמיון אותו בעזרת האלגוריתם מיון הכנסה ניתן ורצוי להיעזר בפונקציה מסעיף א.

עליכם לכתוב main שיקבל 50 מספרים מהמשתמש יכניס אותם למערך בגודל 50 (יש להגדיר עם define) ימיון את המערך וידפיס למסך את המערך הממוין.

השורה האחרונה בפלט חייבת להיות המערך הממוין כאשר האיברים מופרדים ב', ללא רווחים נוספים.

לדוגמא עבור הקלט: 10 9 3 6

הפלט יהיה: 3,6,9,10

עבור הדיבאג מומלץ לקבל קלט מקובץ כמו שראינו בתרגול על לינוקס.

ניתן להניח קלט תקין (אינטגרים חיוביים בלבד) אך אין להניח כי כל המספרים יהיו שונים זה מזה.

מחרוזות 70 נקודות

חיפוש בטקסט

כתבו תוכנית המקבלת כקלט מחרוזות וטקסט ומבצעת את אחת מהאפשרויות הבאות:
א. הדפסת כל השורות בטקסט בהם מופיעה המחרוזת.
ב. הדפסת כל המילים בטקסט הדומות למחרוזות עד כדי השמטת אות אחת מהמילה.
לדוגמא: אם ניתן הטקסט הבא ואני מחפשים את המילה 'cat'

```
this is a text file
looking for the word cat
the program should print also cats
after cats crat and lcat are printed
the program shouldn't print
the word caats
```

התוכנית תדפיס עבור אפשרות ב' את המילים:

```
cat
cats
cats
crat
lcat
```

ועבור אפשרות א' תדפיס:

```
looking for the word cat
the program should print also cats
after cats crat and lcat are printed
```

מכיוון שהמחרוזת 'cat' מופיעה בשורות אלו.

הערה: המילים בטקסט מופרדות באחד מהתווים הבאים: ' ' (space) '\t' (tab) '\n' (newline) וכל שורה מסתיימת בתו '\n'

חלוקה לפונקציות

עליכם לממש את הפונקציות הבאות:

int getline(char s[])

הפונקציה מקבלת מחרוזת (תו אחר תו) מה standard input ושומרת אותה במחרוזת s. הפונקציה תחזיר את כמות התווים שנקלטו בפועל.

יש להגדיר את גודל השורה כקבוע בתוכנית `#define LINE 256`

שימו לב כי שורה תמיד תסתיים בתו '\n'

int getword(char w[])

הפונקציה מקבלת מחרוזות (תו אחר תו) מה standard input ושומרת אותה במחרוזת w. הפונקציה תחזיר את כמות התווים שנקלטה בפועל.

יש להגדיר את גודל המילה כקבוע בתוכנית 30 WORD #define
שימו לב כי מילה תמיד תסתיים באחד התווים ' ', '\t', '\n'

int substring(char * str1, char * str2)

הפונקציה מקבלת שתי מחרוזות str1 ו-str2 ובודקת האם str2 מוכלת ב-str1. (תת מחרוזת לדוגמא : aba היא תת מחרוזת של aagabahj. הפונקציה תחזיר 1 אם כן ו-0 אם לא

int similar (char *s, char *t, int n)

הפונקציה תקבל שני מחרוזות s ו-t ומספר n. הפונקציה תבדוק אם אפשר להגיע מהמחרוזת S למחרוזת t ע"י השמטת n תווים. לדוגמא אם נתון שהמחרוזת S היא "swsystems" והמחרוזת t היא "ssysems" ו n=2 הנתונה תחזיר 1 כי ניתן להשמיט מ S את 't' 'w' וכך לקבל את המחרוזת t. עבור מחרוזות זהות והמספר 0 הנתונה תחזיר 1 רק אם שני המחרוזות זהות.

void print_lines(char * str)

הפונקציה מקבלת את המחרוזת הרצויה לחיפוש, קולטת את שורות הטקסט, ומדפיסה את השורות בהם מופיעה

המחרוזת, תוך שימוש בפונקציות הרלוונטיות שהוגדרו לעיל

void print_similar_words(char * str)

הפונקציה מקבלת את המחרוזת הרצויה לחיפוש, קולטת את מילות הטקסט ומדפיסה את המילים הדומות למחרוזת

החיפוש עד כדי השמטה של אות אחת מהמילים המופיעות בטקסט (כולל מילים זהות למחרוזת החיפוש).

הפונקציה תשתמש בפונקציות הרלוונטיות שהוגדרו לעיל.

שימו לב. החלוקה לפונקציות היא הצעה שנועדה לעזור לכם במימוש. אין חובה לממש את כל הפונקציות עם החתימות הקיימות. הבדיקה תהיה רק בעזרת קלט ופלט.

קלט

את הקלט יש לקבל מקובץ באמצעות redirection (./txtfind < input.txt)
הפורמט של קובץ הקלט ייראה באופן הבא :

בשורה הראשונה תופיע מחרוזת החיפוש (cat) ולאחריה אפשרות החיפוש (a או b)

כאשר a מסמנת את האופציה הראשונה בא יודפסו השורות אשר המילה cat מופיעה בהם וב היא

האופציה השנייה אשר תדפיס את המילים הדומות לcat עד כדי השמטת אות אחת מהמילה.

ניתן להניח קלט תקין. שורה ראשונה תכיל מילה אחת רווח ואות. שורה שנייה ריקה. והחל מהשורה השלישית יהיה הטקסט. אורך הטקסט לא יעלה על 250 שורות.

```
cat a

this is a text file
looking for the word cat
the program should print also cats
and crat and lcat but it shouldn't
print the word caats
```

הגשה

במודל יש להגיש קובץ txt בשם ss_hw3.txt
הקובץ יכול 3 שורות. קישור ל-git שלכם, מספר commit ותעודות זהות המגישים מופרדים ברווח.

על הגיט שלכם להכיל makefile ברמה הראשונה שלו (לא בתתי תיקיות) ולאחר הרצת הפקודה
make all עליכם לקמפל וליצור שני תוכניות הרצה (עדין ברמה הראשונה של הגיט).
isort – שיכיל את התוכנית של החלק הראשון בתרגיל
txtfind - שיכיל את התוכנית של החלק השני.

יועלה למודל סקריפט בדיקת הגשה אשר יבדוק את נכונות קובץ ההגשה שלכם ואת נכונות הגיט
שלכם (בדיקה פשוטה) לצורך הבדיקות האוטומטיות שלנו. יש לבדוק את קובץ ההגשה שלכם
ונכונות הגיט לפני הגשת התרגיל. **סטודנט שלא יתאים את הקוד שלו לבדיקות האוטומטיות יקבל 0.**

יש לפתוח תיקייה חדש ולהכניס לתוכה את קובץ הבדיקות, קובץ ההגשה ושני קבצי דוגמת הקלט
שיעלו למודל. (הקבצים היחידים בתיקייה) ולהריץ את השורה הבאה.

```
./test_submission_hw3.sh ss_hw3.txt
```

אם התוכנית והגיט שלכם תקינים יודפסו תעודות הזהות של המגישים. תופיע לכם תיקיית גיט
חדשה היא תתקמפל ושני התוכניות שהגשתם יורצו עם קלטים פשוטים ופלט התוכניות שלכם יודפס
למסך.

יש לוודא שהתוכנית מתקמפלת ורצה על גבי ubuntu עם gcc אתם רשאים לעבוד עם כל עורך קוד
שנראה לכם לנכון אך **סביבת הבדיקה תהיה ubuntu**