

מכללה אקדמית הדסה

החוג למדעי המחשב

תרגיל #0: תכנות מערכת ומבוא לתכנות מקבילי--

הכרות עם שפת סי (טהורה)

תכנית a: מיזוג של שני קבצים

מיזוג היא פעולה הקולטת שתי סדרות ממוינות, ומצרפת אותן לכדי סדרה ממוינת אחת (בזמן לינארי בגודל הקלט).

לדוגמה: מיזוג של שתי הסדרות (אחת בכל שורה)

5 7 7 12 14 15

6 8 12

ייתן את הסדרה:

5 6 7 7 8 12 12 14 15

כתבו תכנית הקוראת מהמשתמש שמות של שני קובצי קלט, ושם של קובץ פלט.
קובצי הקלט כוללים שמות של סטודנטים וציוניהם, והם ממוינים לפי שם הסטודנט.
שורה בודדת בכל קובץ תראה:

Dana 80 70 90

כלומר לתלמידה Dana יש שלושה ציונים: 80 70 90.

כל שורה בכל קובץ תכיל מחרוזת ראשונה (אחת בודדת) שהינה מזהה הסטודנט (כדוגמת: Dana),
ואחר סדרה באורך כלשהו של מספרים שלמים שמציינים את ציוני הסטודנט.

על התכנית למזג את שני הקבצים, באופן שאם סטודנט מופיע בשניהם אזי תהיה לו שורה יחידה בפלט. השורה תכלול ראשית את ציוניו כפי שהופיעו בקובץ הקלט הראשון, ואחר את הציונים מקובץ הקלט השני.

את התוצאה יש לכתוב על קובץ הפלט, וכן על הפלט הסטנדרטי.

לדוגמה: אם קובצי הקלט הם:

<u>File1</u>	<u>File2</u>
Dana 80 70 90	Rama 70
Rama 60 80	Yosi 1 2 3 4
Sam 6	

אזי קובץ הפלט יהיה:

Dana 80 70 90

Rama 60 80 70

Sam 6

Yosi 1 2 3 4

הערות

- ניתן להניח כי מזהה של תלמיד יהיה מחרוזת בודדת באורך > 50 תווים.
- ניתן להניח כי בסוף השורה האחרונה מצוי התו '\n'.
- ניתן להניח כי קובץ הקלט תקין.
- קראו כל שורה בקלט פעם אחת (ולא יותר).

תכנית b: חלוקת משפטים למילים

הגדירו בתכנית:

```
struct Sentences {  
    char **_data ;  
    int _num_of_sentences ;  
};
```

באמצעות משתנה מהטיפוס הנ"ל נשמור סדרה של משפטים. כל משפט יכלול סדרת מחרוזות, מופרדות זו מזו באמצעות רווח יחיד. בסוף כל משפט מסתיימת שורה.

לדוגמה: משתנה כנ"ל עשוי להכיל את שלושת המשפטים הבאים במרכיב data :

Y	O	S	I		I	S		A		B	O	Y	\0
D	A	N	A	\0									
I		L	O	V	E		Y	O	U	\0			

כתבו תכנית אשר מקבלת באמצעות וקטור הארגומנטים שם של קובץ.

הקובץ יכלול מספר טבעי המציין כמה משפטים כולל הקובץ, ואחר את סדרת המשפטים. לדוגמה:

```
3  
Yosi is a boy  
Dana  
I love you
```

קראו באמצעות סדרת פקודות getline את המשפטים לתוך מערך שתקצו דינאמית. הניחו כי הקלט תקין, וכי אורכו של כל משפט בו חסום ע"י 1000 תווים.

כתבו פונ' המקבלת משתנה מטיפוס ה: struct הנ"ל ומחזירה משתנה מהטיפוס הנ"ל כך שבמשתנה המוחזר כל שורה תכיל מחרוזת אחת בלבד. המחרוזות השנייה ואילך בכל שורה יועברו לשורה חדשה שתוקצה במבנה הנתונים.

לדוגמה: עבור המבנה הנ"ל יוחזר מצביע למבנה בו מרכיב ה- data נראה:

Y	O	S	I	\0
I	S	\0		
A	\0			
B	O	Y	\0	
D	A	N	A	\0
I	\0			
L	O	V	E	\0
Y	O	U	\0	

הציגו את תכנו של מבנה הנתונים שהתקבל בתום ביצוע הפונ'. כל מחרוזת תוצג בשורה נפרדת.

הערות:

א. כדי להקל עליכם, בפונ' פרוק המשפטים למילים, אתם רשאים לרוץ על מבנה הנתונים המועבר, לספור כמה שורות יש להוסיף לו, להקצות מבנה חדש עם מספר השורות המתאים, ועתה להתחיל להקצות את השורות על-פי הצורך. כלומר מותר לעבור פעמיים על מבנה הנתונים.

ב. אורכה של כל שורה יהיה על-סמך אורך המחרוזת השמורה בה.

ג. כאמור, ניתן להניח כי בין כל שתי מחרוזות במבנה הנתונים מפריד רווח יחיד.

ד. נקבע כי מיד אחרי הזנת מספר המשפטים, המשתמש מקיש על Enter, ואת המשפטים הוא מזין בשורה חדשה. לכן אחרי קריאת מספר המשפטים הוסיפו פקודת getch()

הערות:

א. למותר לציין כי יש להקפיד על **כל** כללי הסגנון התכנותי כפי שמיצינו בשנה א', בפרט ובמיוחד בנושא הזיכרון המוקצה דינאמית.

ב. את התכנית יש, כמובן, לקמפל עם gcc (ולא עם g++)

דוגמה לשימוש ב: `getline`

<https://man7.org/linux/man-pages/man3/getline.3.html>

```
#define _GNU_SOURCE
#include <stdio.h>
#include <stdlib.h>
//-----
int main() {

    char *str = NULL ;
    size_t size = 0;

    getline(&str, &size, stdin) ;
    puts(str) ;
    ...
    free(str) ;
```

מכללה אקדמית הדסה

החוג למדעי המחשב

תכנות מערכת ומבוא לתכנות מקבילי – הנחיות להגשת תרגילים

יש להגיש הגשה אלקטרונית של התכניות באתר הקורס ב moodle. שמות קובצי המקור יהיו:

ex0a.c, ex0b.c

במידה והתרגיל כולל כמה תכניות אז הקבצים ייקראו:

ex1a.c, ex1b.c, ...

במידה ותכנית b בתרגיל כולל כמה קבצים אזי הם ייקראו:

ex1b1.c, ex1b2.c, ...

בראש כל קובץ מקור יש לכתוב את שמות ות.ז של המגשים.
יש לצרף להגשה גם קובץ README הכולל:

- i. תיאור קצר (לטובת משתמש) של התכניות המוגשות.
 - ii. האופן המדויק בו יש לקמפל תכניות אלה (כמובן עם gcc)
 - iii. האופן בו יש להריץ את התכניות (ובהמשך גם: הסדר בו יש להריץ כמה תכניות שונות שמשלימות משימה משותפת)
- קובץ ה README יהיה קובץ טקסט פשוט (שנוצר באמצעות editor), ולא קובץ Word או כל פורמט עשיר אחר.

יש לארוז את הקבצים באמצעות פקודת ה- tar של לינוקס.
שם קובץ ה: tar יהיה:

ex1

יש להגיש את כל הקבצים כמות שהם בלי כל תיקיות

תרגיל שלא יוגש כהלכה לא ייבדק, ולא תהיה אפשרות להגשה חוזרת שלו.

תזכורות חשובות:

- i. הקפידו להרוג תהליכים שרצים בלולאה אינסופית. פקודות ה- Shell: ps, kill יעזרו לכם
- ii. הקפידו לצאת מלינוקס בצורה מסודרת לא ע"י לחיצה על ה- X בפינת החלון, אלא ע"י **תפריט: system/logout**
- iii. זכרו: בלינוקס אנו תלויים זה בזה! ואם לא נעבוד בצורה מסודרת כולנו נסבול מאוד!!

תזכורת בעניין tar

פקודת יצירת הקובץ הדחוס אותו יש להגיש היא:

tar czvf **target-file.tgz** README ex1a.c ex1b.c ex1c.c

הקפידו מאוד, מאוד, מאוד ששם הקובץ שאתם יוצרים יופיע ראשון.
תקלה שכיחה היא שסטודנטים כותבים: tar czvf **ex1a.c** ex1b.c ex1c.c **target-file.tgz**,
וכך הורסים לעצמם את ex1a.c שכן לתוכו נכנס הקובץ הדחוס!

מומלץ מאוד גם לשלוח לעצמכם את הקבצים בדוא"ל מעת לעת, על מנת לשמור גיבוי שלהם!

מומלץ לבדוק שהקובץ שיצרתם תקין, וזאת ע"י העתקתו לתיקייה חדשה, ושם ביצוע פקודת החילוץ:
tar -xzf target-file.tgz

אחר בדיקה שכל הקבצים תקינים:

less *

(הפקודה: **tar -tvf target-file** תציג לכם את שמות הקבצים הנכללים בקובץ ה- tar שלכם)