

ב"ה

תרגיל מס' 6 – מבנים ו-ADT (מתרגל אחראי: מור)

התרגיל שווה 25% מציון התרגול

הוראות הגשה

שאלות בנוגע לתרגיל נא להפנות דרך פורום הקורס שנפתח במיוחד לשם כך:

<https://piazza.com/biu.ac.il/fall2017/89110>

אם לא נענתה תשובה תוך 24 שעות, נא לשלוח אלי (מור) מייל עם לינק לדיון הרלוונטי ואענה. המייל הוא: mor.sinay@gmail.com. בכל מייל יש לציין שם, שם משתמש, מס' קורס, וקבוצת תרגול.

- מועד פירסום: 5/1/17
- מועד אחרון להגשה: 19/1/17 23:55
- יש לשלוח את הקבצים באמצעות האתר: <http://help.cs.biu.ac.il/submit.htm>
- לפני חלוף התאריך הנקוב לעיל.
- שם ההגשה של התרגיל: ex6
- יש להקפיד מאוד על כל הוראות עיצוב הקלט והפלט, כמפורט בכל סעיף וסעיף.
- על הפלט להיראות בדיוק כמו בדוגמאות. אין להוסיף או להשמיט רווחים או תווים אחרים ואין להחליף אותיות גדולות בקטנות או להיפך ® אי-הקפדה על פרטים אלה עלול לגרום ירידה משמעותית ביותר בציון התרגיל עד כדי 0. ראו **הזהרות!**
- להזכירכם, העבודה היא אישית. "עבודה משותפת" דינה כהעתקה.
- אין להדפיס למסך שום דבר מעבר למה שנתבקש בתרגיל.
- יש לוודא שהתרגיל מתקמפל ורץ על השרתים באוניברסיטה (u2) ללא שגיאות/אזהרות.
- אתם יכולים לעבוד עם כל עורך טקסטואלי שאתם מעדיפים. להזכירכם – pico – בשרתי linux שבמעבדות; notepad ב-windows ; או בסביבת פיתוח (IDE) – עבור C השתמשו ב-Visual Studio שכולל בתוכו מהדר ל-windows. מומלץ להתנסות בכולם.

הקפידו על כתיבה לפי קובץ ה-Coding-Style שבאתר הקורס!!

יש להשתמש בחומר נלמד עד תרגול 8 (כולל) בלבד

Ex6 – C Program

הנחיות עבור ex6

עליכם להגיש 4 קבצים (אין להגיש פחות או יותר):

1. main.c שבתוכו נמצאת פונקציית ה-main.
2. פונקציה זו היא האפליקציה שמשתמשת בGADT
3. קובץ header שמחצין את מימוש ה-GADT שלכם
3. קובץ c עם מימוש ה-GADT.
- יש לממש את ה GADT עצמאית ולא להשתמש במימוש מהאינטרנט.
4. makefile

עליכם לממש GADT של רשימה ממויינת, שדות ה struct נתונים לבחירתכם.
עליכם להחצין ולממש את חמשת הפונקציות הנ"ל (את שאר הפונקציות שתיצרו אסור להחצין):

1. יצירת רשימה בעל טיפוס כללי
2. מחיקת רשימה
3. חיפוש איבר ברשימה
4. הוספת איבר לרשימה
5. חיבור איבר לאיבר מהרשימה
6. הסרת איבר מהרשימה
7. הדפסת הרשימה
8. הדפסת גודל הרשימה

את המימוש וההחצנה interface (קובץ header) תעשו דרך קבצים קבצי ה c וה h של הרשימה.

עליכם להיצמד למימוש קובץ ה h-המצורף לתרגיל. כמובן, תצטרכו להוסיף הצהרה על פונק' הדפסה ופונקציית גודל רשימה שלא נמצאות בקובץ

האפליקציה שלכם, קרי קובץ ה-main תפעל על פי האלגוריתם הבא:

1. התוכנית תקלוט מספר שלם 0 או 1 (ניתן להניח שאלו הערכים היחידים שיוזנו).
2. אם הקלט הוא 0, יש לקלוט מערך תווים (בגודל המספר מקסימלי של 15), ליצור רשימה ולמקם את הקלט כראש הרשימה.
3. מיון הרשימה יעשה לפי מערך תווים מסודר לקסיקוגרפית (לדוג' abc יהיה לפני bc) אחרת, יש לקלוט ערכי נקודה בפורמט הבא (x,y), שני הקלטים הנ"ל הם מספרים שלמים המופרדים בפסיק וליצור רשימה ממויינת השומרת טיפוס מסוג struct point כפי שהוצג בתרגול ולמקם את הקלט כראש הרשימה.
3. מיון הרשימה לפי נקודות מסודר באופן הבא $d = \text{abs}(x) + \text{abs}(y)$ (לדוגמא (1,2) יהיה לפני (2,3))
4. בצע את הפעולות הבאות (בלולאה אינסופית, עד לקבלת הערך e או הסרת הקודקוד האחרון ברשימה :)
- a. קלוט קוד פעולה כתו בדיד (ניתן להניח קלט תקין)

- b. עבור הקלט a, קלוט מערך (בגודל מקסימלי 15)/נקודה בהתאם לטיפוס הרשימה והוסף לרשימה במקום המתאים.
- c. עבור הקלט s, קלוט מערך (בגודל מקסימלי 15)/נקודה בהתאם לטיפוס הרשימה וחפש ברשימה.
- יש להדפיס TRUE אם הערך קיים ברשימה ו FALSE אחרת . שימו לב כי הקלט עבור חיפוש הנקודה (3,4) הינו (3,4).
- d. עבור הקלט d, הסר איבר מן הרשימה, אם האיבר אינו קיים יש להדפיס FALSE. (אם קיים, אין להדפיס כלום)
- הסרת האיבר היחיד ברשימה מקבילה למחיקת הרשימה, במקרה זה ניתן להתייחס לכך כקלט של אופציה e (זהו סיום התוכנית, לא יהיה קלט נוסף).**
- e. עבור הקלט p, יש להדפיס את הרשימה כפי שיוסבר בהמשך
- f. עבור הקלט l, יש להדפיס את אורך הרשימה
- עבור הקלט t, יש לחבר איבר מהרשימה לאיבר חדש.
- לדוגמא: עבור הרשימה null->(3,4)-(1,2)
1. בהנתן קלט (1,0) (1,2) הרשימה החדשה תהיה null->(3,4)-(2,2)
2. בהנתן קלט (10,0) (1,2) הרשימה החדשה תהיה null->(11,2)-(3,4)
- עבור הרשימה a->ab->null
- i. בהינתן קלט b ab הרשימה החדשה תהיה a->abb->null
- ii. בהינתן קלט c a הרשימה החדשה תהיה ab->ac->null
- אם האיבר שראשון לא נמצא – כלומר האיבר שצריך לחבר אליו, אז לא לעשות כלום.
- g. עבור הקלט e, מחק את הרשימה וצא מן התוכנית.

הערות:

- ניתן להניח שיוכנסו מספרים שלמים/תווים היכן שמצופה לקבל מספרים שלמים/תווים.
- לא יוכנסו מספרים/תווים שלא מצפים להם או שלא מוגדר מה לעשות כאשר הם מתקבלים
- ניתן להניח שלא יוכנס לעץ אותו מערך תווים פעמיים ולא תתבקשו להכניס שתי נקודות שמרחקן זהה. (גם המקרה של ההוספה לאיבר)
- שימו לב שהפעם אתם נמדדים גם על יעילות של מקום וזמן.

הדפסת הרשימה:

יש להדפיס את הרשימה באופן הבא:

עבור המילים a bc d יודפס: (הזחה של מיקום המילה (מתחיל מאפס) * טאב)

```
a
  bc
    d
```

כאשר נוסיף את המילה b יודפס:

```
a
  b
    bc
      d
```

רשימה המכילה את הנקודה (2,3) יודפס:

```
002|003
```

כאשר נוסיף את הנקודה (14,103) יודפס:

```
002|003
    014|103
```

קובץ ה- header שלכם

```
#ifndef GADT_H
#define GADT_H

#include <stdio.h>

//typedef declaration
typedef void* Elm;
typedef void* SLN;
typedef void* HEAD;
typedef enum { SUCESS, OUT_OF_MEM, BAD_ARGS, FAILURE } Result;

HEAD SLCreate(Elm head_val, Elm(*create_elm>(), void(*cpy_elm)(Elm, Elm),
             int(*cmp_elm)(Elm, Elm), void(*free_elm)(Elm),
             void(*print_elm)(Elm), Elm(*add_elm_to_elm)(Elm, Elm));

void SLDestroy(HEAD head);

Result SLAddListElement(HEAD* head, Elm node);

Result SLRemoveElement(HEAD* head, Elm node);

SLN SLFindElement(HEAD head, Elm node);

void SLAddToElement(HEAD* head, Elm toEl, Elm addEl);

#endif
```

ריצה לדוגמא של תוכנית זאת) **באדום מופיעות הדפסות של התוכנית** ובשחור הקלט של המשתמש) דוגמאות נוספות נמצאות בקובץ הפידבק במערכת ההגשה:

```
1
(1,2)
a
(2,3)
p
001|002
      002|003
t
(1,2) (10,5)
p
002|003
      011|007
d
(2,3)
p
011|007
l
1
d
(2,4)
FALSE
e
```

```
0
ab
a
a
a
b
s
a
TRUE
p
a
      ab
      b
t
a c
p
ab
      ac
      b
e
```

בהצלחה!

