

ב"ה

**תרגיל מס' 6 – מבנים, ADT ועצים (מתרגל אחראי: חן)
התרגיל שווה 30% מציון התרגול**

הוראות הגשה

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

<https://piazza.com/biu.ac.il/fall2015/89110/>

אך ורק אם לא נענתה תשובה תוך 24 שעות, נא לשלוח אלי (חן) מייל עם לינק לדיון הרלוונטי ואענה. המייל הוא: chen.hajaj@biu.ac.il. בכל מייל יש לציין שם, שם משתמש, מס' קורס, וקבוצת תרגול.

- מועד פירסום: 07/01/16
- מועד אחרון להגשה: 23:55 21/01/16
- 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 הוזרתם!
- 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.

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

Ex6 – C Program

הנחיות עבור ex6

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

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

השימוש בלולאות בתרגיל זה אסור בהחלט (כל דבר שניתן לעשות בלולאה, ניתן לעשות גם ברקורסיה!) לשם ההבהרה, לא תופיע שום לולאה בתרגיל (בפרט, לא תופענה המילים for, while).

עליכם לממש GADT של עץ חיפוש בינארי, שדות ה- struct נתונים לבחירתכם. עליכם להחזין ולממש את חמשת הפונקציות הנ"ל (את שאר הפונקציות שתיצרו אסור להחזין):

1. יצירת עץ חיפוש בינארי בעל טיפוס כללי
2. מחיקת עץ חיפוש בינארי (נדרשת פונק' רקורסיבית)
3. חיפוש בעץ חיפוש בינארי (נדרשת פונק' רקורסיבית)
4. הוספת קודקוד לעץ חיפוש בינארי (נדרשת פונק' רקורסיבית)
5. הסרת קודקוד מעץ חיפוש בינארי
6. הדפסת קודקודי העץ (יינתן בונס של 5 נק' על מימוש רקורסיבי)

את המימוש וההחצנה דרך interface (קובץ header), תעשו דרך קבצים קבצי ה- c וה- h של העץ. עליכם להיצמד למימוש קובץ ה- h המופיע במצגת. כמובן, תצטרכו להוסיף הצהרה על פונק' ההדפסה שלא נמצאת במצגת.

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

1. התכנית תקלוט מספר שלם – 0 או 1 (ניתן להניח שאלו הערכים היחידים שיוזנו).
2. אם הקלט הוא 0, יש לקלוט מספר שלם וליצור עץ חיפוש בינארי השומר טיפוס int ולמקם את הקלט כשורש העץ.
3. אחרת, יש לקלוט ערכי נקודה בפורמט הבא (x,y) (שני הקלטים הם מספרים שלמים המופרדים בפסיק), ליצור עץ חיפוש בינארי השומר טיפוס struct point (כפי שהוצג בתרגול) ולמקם את הקלט כשורש העץ. עץ חיפוש בינארי של נקודות מסודר על פי המרחק של כל נקודה מראשית הצירים. מרחק זה יש לחשב באמצעות משפט פיתגורס. דוגמא לערך נקודה (4,5).
4. בצע את הפעולות הבאות (ברקורסיה אינסופית, עד לקבלת הערך e או הסרת הקודקוד האחרון בעץ):

- a. קלוט קוד פעולה כתו בדיד (ניתן להניח קלט תקין)
 - b. עבור הקלט a, קלוט מספר/נקודה בהתאם לטיפוס העץ והוסף לעץ במקום המתאים.
 - c. עבור הקלט s, קלוט מספר/נקודה בהתאם לטיפוס העץ וחפש בעץ, יש להדפיס TRUE אם הערך קיים בעץ ו-FALSE אחרת. הקלט עבור חיפוש הנקודה (3,4) הינו (3,4).
 - d. עבור הקלט d, הסר קודקוד מן העץ. אם הקודקוד אינו קיים יש להדפיס FALSE.
- יש לממש את המחיקה בדיוק כפי שמופיע בקובץ ההסבר הצורף בנוגע ללוגיקת הסרה.
- הסרת הקודקוד היחיד בעץ מקבילה למחיקת העץ, ניתן להתייחס לכך כקלט של אופציה e (זהו סיום התוכנית, לא יהיה קלט נוסף).
- e. עבור הקלט p, יש להדפיס את העץ על פי ההסבר שמופיע בהמשך.
 - f. עבור הקלט e, מחק את העץ וצא מן התוכנית.

הערות:

1. ניתן להניח שיוכנסו מספרים שלמים/תווים היכן שמצופה לקבל מספרים שלמים/תווים.
2. לא יוכנסו מספרים/תווים שלא מצפים להם או שלא מוגדר מה לעשות כאשר הם מתקבלים.
3. ניתן להניח שלא יוכנס לעץ אותו מספר פעמיים ולא תתבקשו להכניס שתי נקודות שמרחקן מראשית הצירים זהה.
4. שימו לב שהמימוש המתבקש הוא לא יעיל מבחינת המקום שהוא תופס (בשל השימוש ברקורסיה), אתם לא נמדדים על יעילות מקום וזמן.

הדפסת עץ:

ניתן להניח שלשם הדפסת ערך כל קודקוד בעץ מספיק להשתמש בחמישה תווים.
 עבור עץ מספרים שלמים – הטווח הינו 9999:99999-
 עבור עץ נקודות – הטווח של ערך כל נקודה הוא 9:99-
 יש להדפיס את העץ על מנת שנוכל לראות את המבנה שלו, כל ריווח בהדפסה משמעו תו רווח.

עץ של מספרים שלמים

למשל העץ המכיל את הערך 2- יודפס כ:

-0002

העץ הנוצר על ידי רצף הערכים הבא (משמאל לימין):

2,1,3

יודפס כך:

00002
00001 00003

העץ הנוצר על ידי רצף הערכים הבא (משמאל לימין):

1,3

יודפס כך:

00001
00003

כאשר נוסף את הערך 4, העץ יודפס כך:

00001
00003
00004

כאשר נוסף את הערך 2, העץ יודפס כך:

00001
00003
00002 00004

עץ של נקודות

העץ המכיל את הנקודה (2,3) יודפס כ:

02|03

כאשר נוסף את הנקודה (3,4), העץ יודפס כך:

02|03
03|04

כאשר נוסף את הנקודה (1,2), העץ יודפס כך:

02|03
01|02 03|04

כאשר נמחק את נקודה (2,3), שורש העץ, העץ יודפס כך:

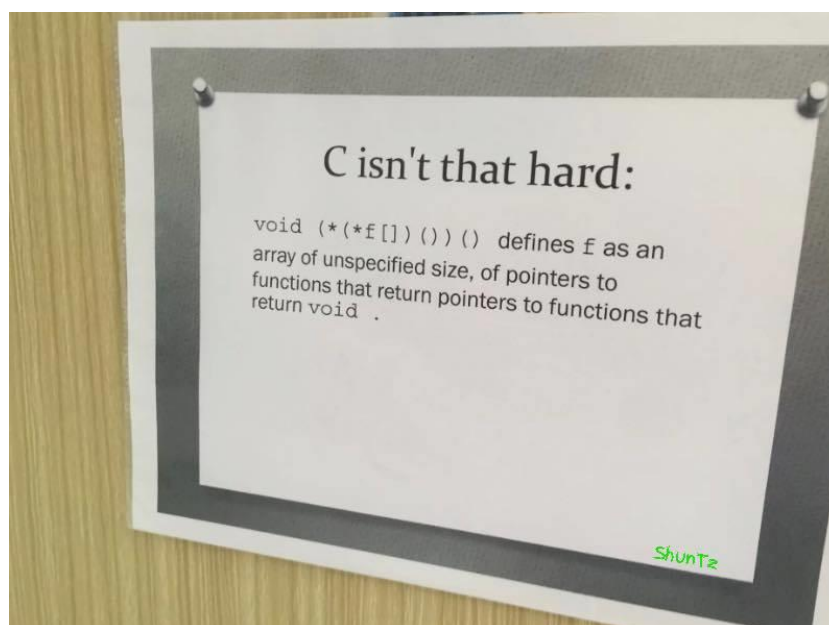
03|04
01|02

הלוגיקה: כל ערך מיוצג על ידי חמישה תווים וההזחה היא לפי מבנה העץ.
 שימו לב שמבנה העץ תלוי בסדר הזנת הערכים.

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
01|02
      02|03
d
(1,2)
p
02|03
d
(2,4)
FALSE
e
```

```
0
1
a
2
a
3
s
3
TRUE
p
00001
      00002
            00003
e
```



בהצלחה!

