תרגיל בית 3 (מעשי)

בתרגיל זה נרצה לממש מבנה נתונים שתומך בפעולות הבאות. כל המספרים הם מספרים טבעיים חיוביים בלבד .

- **Insert(k)** - הוסף רשומה חדשה בעלת מפתח k, אם כבר קיימת כזו אז אל תעשה דבר .
- **Delete(k)** - הוצא רשומה קיימת בעת מפתח k ושחרר את הזיכרון שהוקצה עבורה. אם לא קיימת רשומה כזו אל תעשה דבר .
- **DeleteAll()** – מחק את כל הרשומות הקיימות כעת במבנה ושחרר את הזכרון שהוקצה עבורן.

בנוסף נרצה לתמוך בשאילתות הבאות :

- **Search(k)** - מצא והחזר את הצומת בעל מפתח (kאם יש כזה .)
- **Rank(k)** - החזר את מספר האיברים שקטנים או שווים ל kונמצאים כרגע במבנה .
- **Select(k)** - החזר את הרשומה שה Rank של המפתח שלה הוא k (כלומר את הרשומה בעלת המפתח ה k בגודלו). אם k>n אז יש להחזיר את התשובה עבור k=n ואם k<1 אז יש להחזיר את התשובה עבור k=1 .
- **NextMissing()** - מצא והחזר את המספר הטבעי הקטן ביותר שגדול מהמפתח המינימלי במבנה, אך שלא נמצא במבנה. אם המבנה ריק, יש להחזיר 1 .
(מומלץ קודם כל לתכנן אלגוריתם יעיל, כחלק מתרגיל הבית היבש, לבעיה ורק לאחר מכן לתכנת אותה)

סיבוכיות הזמן הנדרשת לכל אחת מהשאילתות : $O(\log(n))$ (מלבד *DeleteAll*)

כדי לפתור בעיה זו עליכם לממש עץ דרגות מסוג AVL .

<u>התרגיל</u>

בקבצי התרגיל, מצורף לכם קובץ avl.h ובו נמצאת הגדרת הרשומה (צומת בעץ) והחתימות של הפונקציות המתאימות לפעולות ושאילתות שהגדרנו. קראו את הקובץ היטב והבינו מה הם הפרמטרים וערכי החזרה של כל אחת מהפונקציות. בנוסף, מצורף לכם קובץ avl.c כתוב חלקית, שבו עליכם לממש את הפונקציות הללו (ועוד פונקציות עזר כרצונכם). קראו את ההערות גם בקובץ זה ועקבו אחריהן. בשום מקרה אין להשתמש ברשומה אחרת מזו שנמצאת בקובץ avl.h ואין לשנות אותו.

.**קומפילציה**

לפני שהתחלתם לתכנת, וודאו שבהגדרות סביבת העבודה שלכם (בין אם Visual Studioאו Code::Blocks) הקומפיילר הוא GCC ושהדגלים הבאים מודלקים :

- -Wall
- -Wextra
- -pedantic-errors

בדיקת התרגיל

בקובץ avl.c מצויות כמה פונקציות בדיקה בסיסיות. הן בודקות שהעץ שמתקבל הוא אכן עץ AVL תקין ושכל הפעולות והשאילתות עובדות כמו שצריך. יהיו פונקציות בדיקה נוספות שעליהן התוכנית שלכם תיבדק אך הן לא יפורסמו, כך שמומלץ לכתוב פונקציות בדיקה משלכם כדי לוודא שאין בתוכנית באגים .

ניקוד

- תוכנית שאינה מתקמפלת או שקורסת לא תקבל נקודות.
- תוכנית שרצה בזמן של יותר מ 3 שניות לא תקבל נקודות. הבדיקות יתבצעו במחשב עם מעבד i5 ו 8GB זיכרון .RAM
- תוכנית שנכשלת בביצוע הכנסות ומחיקות או שהעץ שמתקבל אינו AVL לא תקבל נקודות .
- דליפת זיכרון תגרום להפחתה של 10 נקודות .

Insert+Delete+DeleteAll+Search = 56 points.

Insert+Delete+DeleteAll+Search+Rank = 70 points.

Insert+Delete+DeleteAll+Search+Rank +Select = 85 points.

Insert+Delete+DeleteAll+Search+Rank+Select+NextMissing = 100 pt

הוראות כתיבה והגשה

- אין להשתמש בספריות נוספות מלבד stdio ו stdlib .
- אין לשנות את הגדרת הרשומה או להשתמש בכל רשומה אחרת מלבד זאת שמופיעה בקובץ avl.h .
- מלאו בראש הקובץ avl.c את השמות ואת תעודות הזהות שלכם בהתאם (ראו תמונה)

```
/****
  Student1 name: ----- -----
  Student2 name: ----- -----

  Student1 ID: ----- -----
  Student2 ID: ----- -----
****/
```

- יש להכין קובץ pdf בשם Documentation.pdf המתאר בקצרה כיצד פתרתם כל שאילתא
- יש להגיד אך ורק את avl.c נקי מפונקציות בדיקה/ פונקציית main ואת הקובץ Documentation.pdf בתוך קובץ ZIP/RAR עם הת"ז של המגישים בצורה הזו : HW3_ID1_ID2.Zip. כאשר במקום ID1 וID2, הת"ז שלכם.
- הקוד שהוגש צריך להיות מתועד ונקי .
- **מטרת התרגיל היא ללמוד ולתרגל את החומר מהכיתה, העתקות מכל סוג (חברים, קוד מהאינטרנט וכו') יטופלו בחומרה!**

טיפים מומלצים (לא חובה)

- התחילו במימוש הפונקציות הבאות. אלו הן הפונקציות הפשוטות ביותר למימוש :

new_avl_node, delete_avl_tree, avl_search, avl_rank, avl_select

  - ממשו את הפונקציות כך שיהיו רקורסיביות. אם הפרמטרים לא מתאימים לרקורסיה אז הוסיפו פונקציית עזר רקורסיבית .
  - כתבו פונקציות עזר רבות. הקפידו שכל פונקציה תהיה קטנה מ 25שורות .
  - השתמשו בכך שהמצביעים לבנים של צומת נמצאים במערך, כדי לחסוך בקוד היכן שיש סימטריה בין ימין ושמאל .

בונוסים –

3 התרגילים הכי מהירים (זמן ריצה של התוכנית) על הטסטים, יזכו לבונוס לציון של התרגיל (ייתכן ציון מעל 100). הבונוס יינתן רק לתרגילים שמימשו הכל.

מקום 1 – בונוס של 20 נק'

מקום 2 – בונוס של 15 נק'

מקום 3 – בונוס של 10 נק'

בהצלחה !