**עבודת בית 2 - מעשי**

מתרגלת אחראית: גנאדי קוגן

תאריך פרסום: 21/11/21

תאריך הגשה: 6/12/21 עד שעה 23.59

**הנחיות כלליות:**

* קראו היטב את השאלות.
* **ניתן להגיש את העבודה בזוגות או ביחידים.**
* הגשת העבודה תהיה דרך אתר הקורס במודל.
* יש לכתוב תכנית המכילה ארבע קבצים (קובץ הגדרות - קובץ עם סיומת .h, קובץ המכיל כל גופי פונקציות, קובץ ראשי – main, קובץ readme – הסבר על הפעלת תוכנית ). העבודה תמומש בשפת תכנות C.
* יש לכווץ כל קבצי התוכנית לקובץ zip.
* שם הקובץ שיוגש למערכת ההגשה יהיה מורכב מת"ז של המגיש/ים.

לדוגמה:

111111111.zip - עבור הגשה ביחיד

111111111\_22222222.zip - עבור הגשה בזוג

* לפני ההגשה יש להשתייך לקבוצת ההגשה. במקרה של הגשה בזוגות, רק אחד מבני הזוג יגיש את העבודה במודל**.**
* חובה לתעד את הקוד.
* במידה ומצאתם טעויות בעבודה או בקשות להבהרות, אנא שלחו מייל ל- genadko@ac.sce.sc.il
* איחור במועד ההגשה יגרור הורדה של ציון, 5 נק' לכל יום איחור או חלק ממנו.  
  בכל מקרה לא יהיה ניתן להגיש מעבר ל-3 ימי איחור ממועד ההגשה המקורי.  
  במקרים חריגים בלבד יש לפנות למרצה כדי לקבל אישור על הגשה באיחור.
* שאלות לגבי העבודה יש לשאול בפורום באתר הקורס ("מודל") או בשעות קבלה של המתרגל האחראי **בלבד**. ***אין לשלוח שאלות במייל המתרגלות או המרצה.***
* להזכירכם: יש לשמור על הגינות אקדמית!

**כללי:**

* עליהם לכתוב פונקציות הבאות המטפלות בעץ בינארי:

**שאלה 1 (14 נקודות):**

כתבו פונקציה **int isEmpty(node\* root)** המקבלת מצביע לשורש של עץ בינארי, ובודקת האם העץ הוא ריק.

הפונקציה מחזירה ערך 1 אם עץ ריק, ו-0 במקרה אחרת.

**שאלה 2 (15 נקודות):**

כתבו פונקציה **רקורסיבית** **void printLeaves(node\* root)** המקבלת מצביע לשורש העץ ומדפיסה את כל עלים בעץ.

**יש להדפיס את העלים לפי סדר הופעתם מימין לשמאל בעץ עם רווח יחיד ביניהם.**

למשל, עבור עץ שבציור הפונקציה תדפיס (מימין לשמאל): 9 40 30 42. הבהרה: 9 הוא עלה הכי ימני, לכן יודפס ראשון. 40 הוא עלה הבא בסדר מימין לשמאל וכך הלאה.

הערה: במידה ועץ ריק, פונקציה לא תבצע דבר.



דוגמה להדפסה: 42 30 40 9

**שאלה 3 (14 נקודות):**

כתבו פונקציה **int contains(node\* root, int key)** המקבלת מצביע לשורש העץ ומתפתח key ובודקת האם מפתח key נמצא בעץ בינארי.

* אם key מופיע בעץ, הפונקציה תחזיר ערך 1
* אם key לא מופיע בעץ – הפונקציה תחזיר 0.

למשל, עבור עץ מסעיף הקודם:

* עבור ערך חיפוש 44 הפונקציה תחזיר 1
* עבור ערך 12 הפונקציה תחזיר 0.

**שאלה 4 (15 נקודות(:**

כתבו פונקציה **int countNodes(node\* root)** המקבלת מצביע לשורש העץ ומחשבת כמות הכוללת של צמתים הפנימיים בעץ.

לדוגמה: עבור העץ משאלה 2 הפונקציה מחזירה 8.

**שאלה 5 (14 נקודות):**

כתבו פונקציה **int sumOfInternalKeys(node\* root)** המחשבת ומחזירה סכום ערכי צמתים הפנימיים של עץ בינארי.

לדוגמה: עבור העץ משאלה 2 או עבור העץ משאלה 6 (בשתי שאלות הוצג אותו עץ) הפונקציה מחזירה 198.

הערה: במידה ועץ ריק, פונקציה לא תבצע דבר.

**שאלה 6 (14 נקודות):**

כתבו פונקציה **int levelStatistics(node\* root, int level)** המדפיסה מפתחות של כל הצמתים ששייכים לרמה level **וגם** מחשבת ומחזירה את מספר הצמתים ברמה.

לדוגמה עבור העץ שמשאל:

* + לרמה 0 – הפונקציה מדפיסה 20 ומספר צמתים 1
  + לרמה 2 – הפונקציה מדפיסה מפתחות (משמאל לימין):

30 44 17 10 ומספר צמתים 4

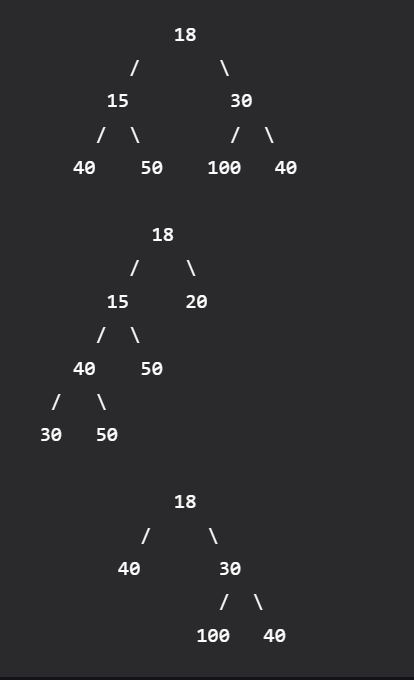
**תחילה יש להדפיס את המפתחות משמאל לימין עם רווח יחיד ביניהם**

**לאחר מכן, בשורה הבאה יש להדפיס את מספר הצמתים ברמה זו.**

**שימו לב:** יש לבדוק תקינות פרמטרים שהפונקציה מקבלת, כלומר אם level אינו חוקי, יש להדפיס הודעת שגיאה “Not a valid level”.

**שאלה 7 (14 נקודות):**

כתבו פונקציה **int** **isFull(node\* root)** הבודקת האם עץ הוא עץ מלא (Full Binary Tree). אם העץ מלא, הפונקציה תחזיר 1, אחרת היא תחזיר 0. הערה: עץ ריק הוא גם עץ מלא. לדוגמה:



**עבודה נעימה!**