**בשאלות על assert:**

* Assert היא macro function המיועדת לדיבוג
* לחשוב מה קורה אם אני מכבה את הדגל ואין לי assert? האם התוכנית עדיין עובדת?
* למה יש צורך לכבות את הassert?
* "נניח שאני מכבה את הassert – זה עדיין טוב או לא טוב?"
* לעבור על השקפים, להבין מה התפקיד העקרוני של assert. זאת לא פונקציה רגילה – ניתן לכבות אותה.
* האם זה שגיאה בקוד, שגיאה בנתונים או אחר.

**בפונקצייה של היבש**

* איפה עושים את הfree למחרוזות? לא צריך לקרוא לפונקציה מתוך הprintf, ליצור משתנה ולשחרר אחכ.
* **מותר** להשתמש ב-realloc. זה לא מה שהוא תכלס רצה אבל הם יקבלו את זה.

**בחלק הרטוב:**

* **על הreSetים**:
* אנחנו עובדים עם טיפוס נתונים. מקובל כשמתחילים לעבוד עם טיפוס נתונים, יש פעולת create וכשמסיים יש את פעולת destroy. הצמד הזה רלוונטי ליחידה **אחת** חדשה.
* טיפוס הנתנוים העיקרי הוא room\_system. ממומש ע"י struct עם שדות שאנחנו צריכים לממש.
* בתוך הסטראקט של הסיסטם רוצים שיהיה שם מערך של challenges (יודעים מראש כמה challenges יש לכן גודל המערך ידוע מראש ו**קבוע**). המערך מכיל מתוכו המון אתגרים. אתגר הינו איבר במערך.
* יוצרים את המערך בגודלו, ואז לכל אתגר במערך נעשה create עם איתחול. בכדי לשחרר את המערך נעשה reset בכדי להשאיר את החלק בתוך הchallenge ריק ואז לשחרר את המערך. => הdestroy הוא על הsystem. Reset לא עושה free לאתגר כאתגר. הוא משחרר את השדות שזקוקים לשחרור.
* לשים לב מה קורה בreset על room – לא לשחרר מידע לא רלוונטי (מידע על אתגר שלא קשור בהכרח לחדר עצמו)
* קובץ עם include בתוך struct (?!)
* האם reset מכין את האיבר לשימוש חוזר? בעיקרון, יועד לשחרור המערכים בשביל הdestroy על הsystem.
* Reset ברמת האתגר, הנוגע לרמת קושי – אפשר לאתחל ל-0 פשוט.
* **Initialize:**
* אפשר להחליט האם לעשות malloc? – משאיר לנו להחליט.
* נשים לב שהכול מתחיל בcreate על הsystem שמגדיר את כלל המערכת (כמה חדרים, כמה אתגרים וכו') ולכן לשים לב לכתוב באופן **מודולרי** את כלל הטיפוסים במערכת (מערך של אתגרים, ואז לאתחל כל איבר בפני עצמו).
* על room – הוא מכיל מערך activity. האם אתחול שלו זה מערך בגודל קבוע? כן. כמות הactivity קבוע. השינוי בזמן הריצה זה על הvisitor.
* נשים לב שבסיסטם יש רשימה **דינאמית** של מבקרים – נממש ברשימה מקושרת (מצגת של תרגול 4)
* הביקור של מבקר נספר בעת הכניסה!
* Char\*\* = מערך של **מצביעים**
* **עידכוני זמנים:**
* צריך לבדוק האם הזמנים חוקיים – צריך "מישהו" שידע מתי התרחשה הפעולה האחרונה במערכת. רמז: מדובר על גורם "על" שישמור את הזמן האחרון כל פעם, בכדי שנוכל להשוות מולו. (משתנה מהסיסטם כנראה)
* בסופו של דבר הגורם שרואה בראיית רוחב את הפעולות והסדר הכרונולוגי של הזמנים זה חייב להיות ברמת הסיסטם.
* האם ניתן שמבקר נכנס ויוצא ב0 זמן? אם יש זמן 0 מבחינתנו אין מבקר. נניח שזמן היציאה **גדול** מזמן הכניסה. לא יהיה טסט שאומר אחרת.

**בקשר למיילים (ציטוט של ישראל):**

* אין מצב של 100% היענות למיילים.
* ננסה ליצור עוד סדנה בפורום של מיועד לתרגיל לפני ההגשה. נעדכן.
* יכול להיות שידחו את פרסום התרגיל הבא בהתאם לתאריכים החדשים.
* נבדוק את נושא הFAQ