

מעבדה מתקדמת בשפת C

סדר פעולות בדיקת קובץ H

הנחיות אלו נכתבו בדם (בצורת נקודות שירדו סתם), ונותנות תיאור חלקי של סדר הפעולות שמתבצע בעת בדיקת התרגילים. מומלץ מאוד לסטודנטים לבצע אותן בעצמם לפני ההגשה, וכך להימנע מהורדת נקודות מיותרת.

1. עוד לפני שפתחנו את הקובץ - האם שם הקובץ הגיוני? האם מתכנת סביר שמכיר את דרישות התרגיל, אבל רואה רק את שמות הקבצים בתיקיה, יכול להעריך במידה סבירה אילו פונקציות נמצאות בקובץ הזה, ואילו נמצאות בקבצים האחרים?
 - 1.1. בהקשר הזה, שמות שלא מאפשרים להסיק כלום לגבי תוכן של קובץ, הם עילה להורדת נקודות משמעותית. למשל `functions.h` – מה אנחנו לומדים על תוכן הקובץ מהשם הזה? נשמע שהדבר היחיד שאפשר להסיק זה שהקובץ מכיל פונקציות, אבל זה בעצם נכון (כמעט) לכל קובץ, אז לא קיבלנו שום אינפורמציה על תוכן הקובץ.
 2. פתחנו את הקובץ. האם הוא מתחיל ב-`include guard`? אם לא, הורדת נקודות אוטומטית.
 - 2.1. האם ה-`include guard` הוא אכן תרגום סביר של שם הקובץ לאותיות גדולות?
 3. בהנחה שלקובץ ה-H יש "בן זוג", קובץ C באותו שם - לכל פונקציה שמוצהרת בקובץ ה-H:
 - 3.1. האם הפונקציה ממומשת בקובץ ה-C הנ"ל?
 - 3.2. האם הפונקציה נקראת מקובץ C אחר, כלומר הפונקציה היא חיצונית, ולא פנימית?
 - 3.3. חשוב שלכל פונקציה ומשתנה יהיה שם משמעותי שמעיד על תפקידה בקוד. ובמקרה הזה, מכיוון שהפונקציה היא חיצונית, חשוב אף יותר שהשם שלה יהיה משמעותי. האם אותו מתכנת סביר שמכיר את דרישות התרגיל אבל לא את הקוד, יכול להסיק במידה סבירה מה הפונקציה עושה, מבלי לקרוא את המימוש שלה בקובץ ה-C?
 - 3.4. אם לקובץ ה-H אין קובץ C מתאים, נצפה שלא יהיו בו פונקציות שמוצהרות. המקרה הנפוץ בו זה קורה הוא קובץ "הגדרות שונות", שנהוג לקרוא לו `common.h`. המצב הזה לרוב לא אמור לקרות במקרים אחרים.
 4. לכל טיפוס נתונים שמוגדר בקובץ ה-H:
 - 4.1. האם טיפוס הנתונים הזה מופיע בקובץ C מלבד בן הזוג של קובץ ה-H? כלומר, האם הטיפוס הוא חיצוני, או פנימי למודול?
 - 4.2. האם השם של טיפוס הנתונים משמעותי, ומעיד על תפקידו בקוד?
 - 4.3. האם טיפוס הנתונים הזה שייך לוגית לקובץ ה-H הזה? כלומר, האם רוב הפונקציות שמתעסקות בטיפוס הנתונים הזה נמצאות בקובץ ה-C המתאים?
 - 4.3.1. אם נתתם דעתכם על הסוגיה אבל קשה לכם להכריע, רישמו הערה בקוד שמבהירה קודם כל שבכלל חשבתם על הנושא, ואח"כ מסבירה את השיקול מאחורי הבחירה שלכם.

- 4.4. בהזדמנות זו, אם יש צורך בטיפוס נתונים בוליאני, הדבר הנכון לעשות הוא להשתמש בקובץ הספריה הסטנדרטית `stdbool.h` שמגדיר טיפוס כזה, ולא להגדיר אותו בעצמכם. (ומכיוון שזה רשום במסמך הזה, זו עילה להורדת נקודות אם הגדרתם אותו בעצמכם).
5. לכל דבר אחר שמוגדר בקובץ, בעיקר קבועים:
- 5.1. האם השם משמעותי מספיק בשביל שאותו מתכנת מיתולוגי יוכל להבין במה מדובר? למשל, אם מדובר בגודל או אורך מקסימלי, האם ברור "גודל או אורך של מה"?
- 5.2. האם הקבוע (או הדבר האחר) נמצא בשימוש בקבצי C מלבד בן הזוג? כלומר, האם הוא חיצוני או פנימי למודול?
6. האם איזשהו חלק מהקובץ הוא `copy-paste` ממקום אחר בקוד, כלומר יש שכפול קוד או הגדרות?
7. אם קובץ ה-H מכיל `include` לקבצי H אחרים, כולל קבצי H מהספריה הסטנדרטית כגון `stdlib.h`: האם ה-`includes` הללו אכן נדרשים? האם קובץ ה-H משתמש בטיפוסים, קבועים, או כל דבר אחר שמגיע דרך ה-`include`? אם לא, ה-`include` מיותר ויש למחוק אותו.
- 7.1. שימו לב: קובץ ה-H עצמו צריך להשתמש בטיפוסים, קבועים, או כל דבר אחר שמגיע דרך ה-`include`. אם קובץ ה-C בן הזוג שלו משתמש בפונקציות שמוגדרות בספריה הסטנדרטית, הפתרון הוא שקובץ ה-C יעשה `include` לקובץ מהספריה הסטנדרטית, ואילו קובץ ה-H לא יעשה `include` לקובץ מהספריה הסטנדרטית. למשל, אם קובץ ה-C הרלבנטי מדפיס משהו למסך ולכן הוא צריך `include` ל-`stdio.h`, זהו לרוב פרט מימוש של קובץ ה-C, ולכן ה-`include` צריך להימצא בקובץ ה-C ולא בקובץ ה-H.

בהצלחה!