עבודה מסכמת – מערכות הפעלה

שאלה 1:

char globBuf[65536]; */\* 1. Where is allocated? \*/*

1. המשתנה globBuf נמצא ב uninitialized data segment (bss).

לפי מבנה הזכרון של תכניות בשפת c, משתנים גלובליים יכולים להימצא או ב uninitialized data segment או ב initialized data segment כאשר המשתנה ממוקם בחלק הראשון אם המתכנת לא איתחל אותו ובחלק השני כאשר המתכנת איתחל את המשתנה.

מאחר ו globBuf לא אותחל אז הוא ממוקם ב uninitialized data segment.

\*מופיעים בסוף הקובץ צילומי מסך שמאשרים קביעה זו.

int primes[] = { 2, 3, 5, 7 }; */\* 2. Where is allocated? \*/*

1. המשתנה primes נמצא ב initialized data segment.

לפי מבנה הזכרון של תכניות בשפת c, משתנים גלובליים יכולים להימצא או ב uninitialized data segment או ב initialized data segment כאשר המשתנה ממוקם בחלק הראשון אם המתכנת לא איתחל אותו ובחלק השני כאשר המתכנת איתחל את המשתנה.

מאחר ו primes אותחל אז הוא ממוקם ב initialized data segment.

\*מופיעים בסוף הקובץ צילומי מסך שמאשרים קביעה זו.

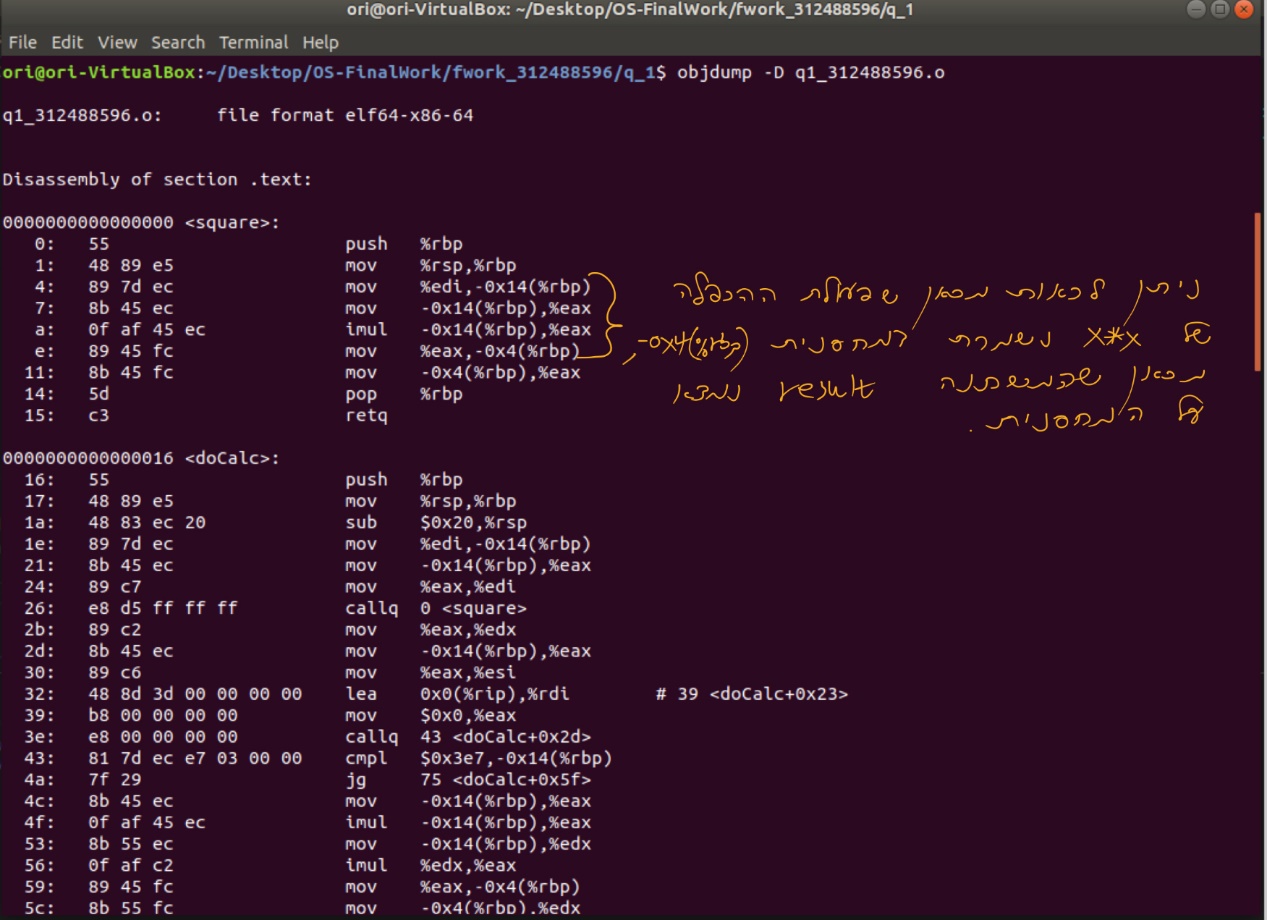
**static** int square(int x) */\* 3. Where is allocated? \*/*

1. הפונקציה square נמצא ב text segment.

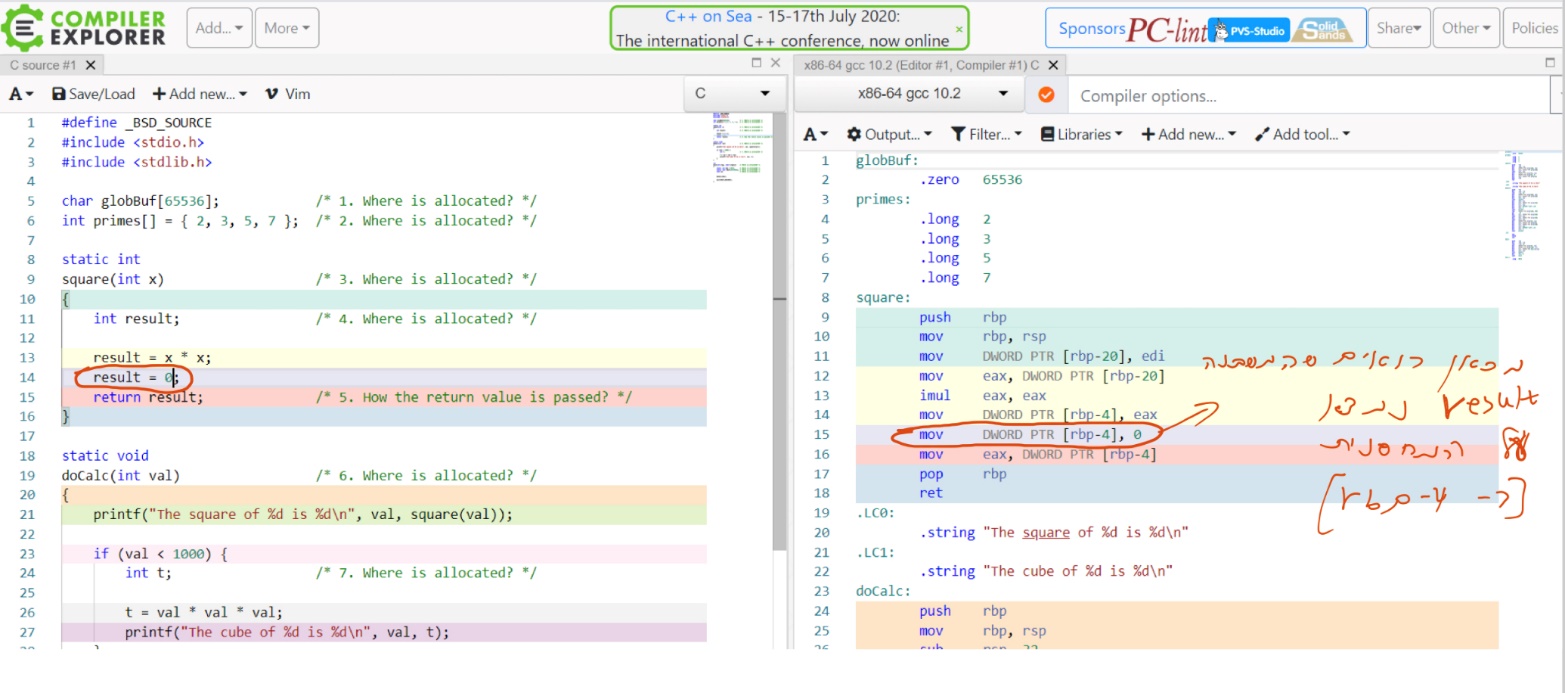
int result; */\* 4. Where is allocated? \*/*

1. המשתנה result נמצא ב stack.

ניתן לראות מהצילום הבא שהשורה result=x\*x מתבצעת והתוצאה נשמרת בסופו של דבר על המחסנית בהיסט של 4- מתחילת ה frame של המחסנית של הפונקציה ולכן מוכח מכאן שהמשתנה result נמצא על המחסנית.



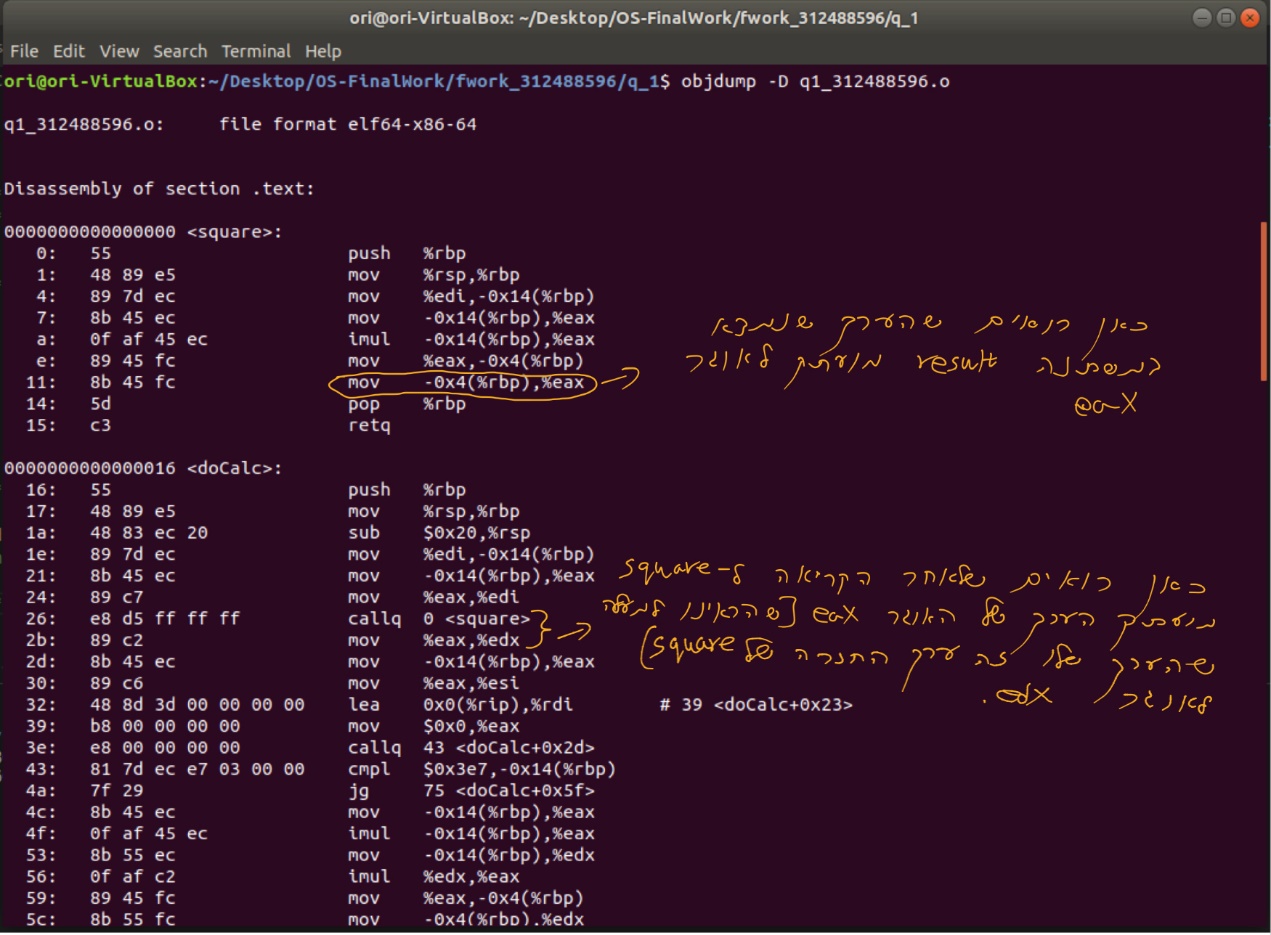
בצילום הבא ניתן לראות את קוד האסמבלי של התכנית (מהאתר godbolt) עם הוספה של שורה 14 (result = 0), ניתן לראות מקוד האסמבלי של שורה זו שהמשתנה result נמצא על המחסנית במיקום rbp-4.

****

**return** result; */\* 5. How the return value is passed? \*/*

1. ערך החזרה מועבר על-ידי רגיסטר.

בצילום הבא של קוד האסמבלי של התכנית ניתן לראות שהערך שנמצא במשתנה result (בפונקציה square) מועתק לאוגר eax, כמו כן בקוד האסמבלי של הפונקציה doCalc ניתן לראות שלאחר הקריאה לפונקציה square מועתק הערך שנמצא באוגר eax שכמו שאמרנו בהתחלה הוא מכיל את הערך של result שזה ערך החזרה של הפונקציה square.



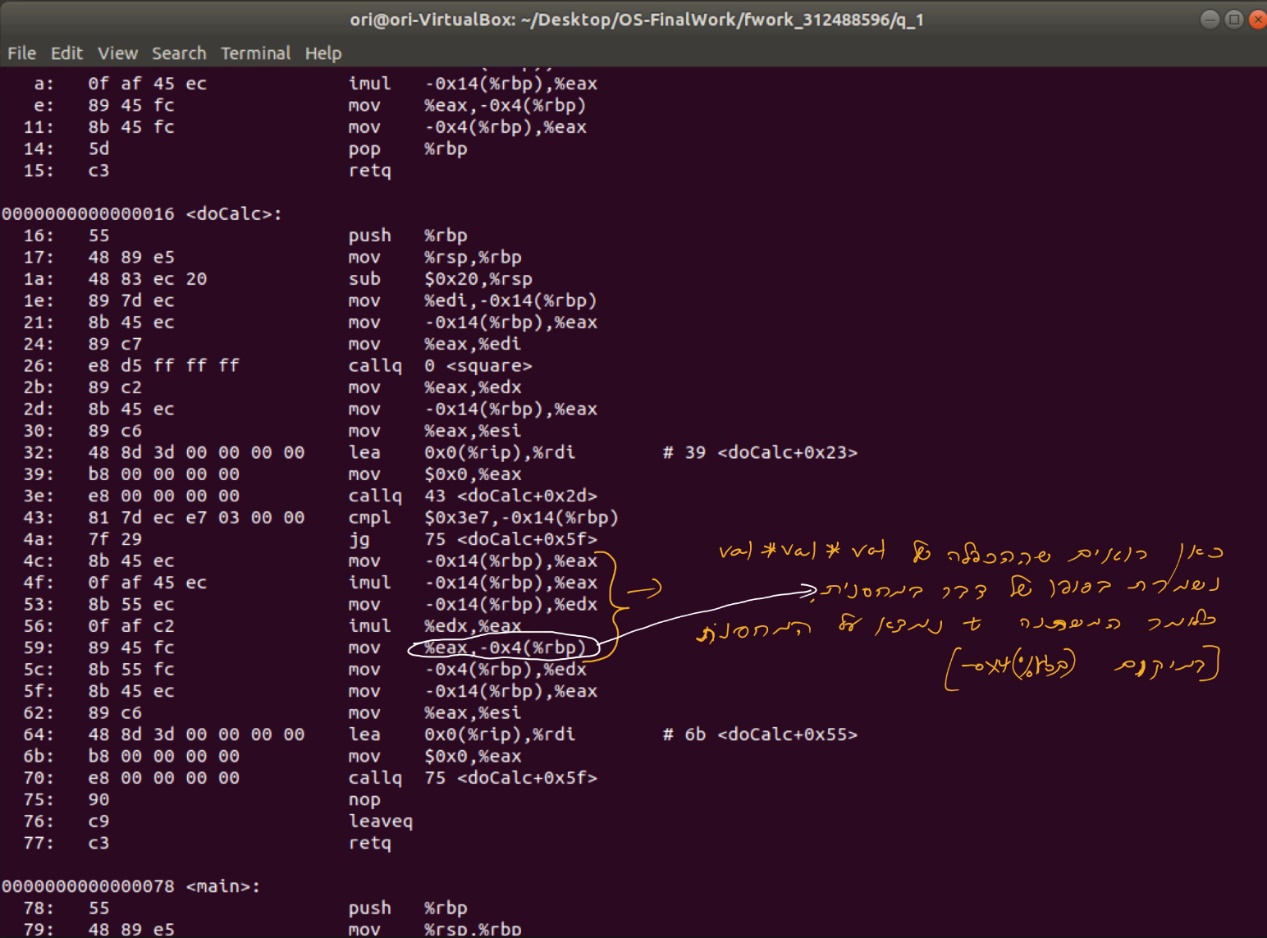
**static** void doCalc(int val) */\* 6. Where is allocated? \*/*

1. הפונקציה doCalc נמצא ב text segment.

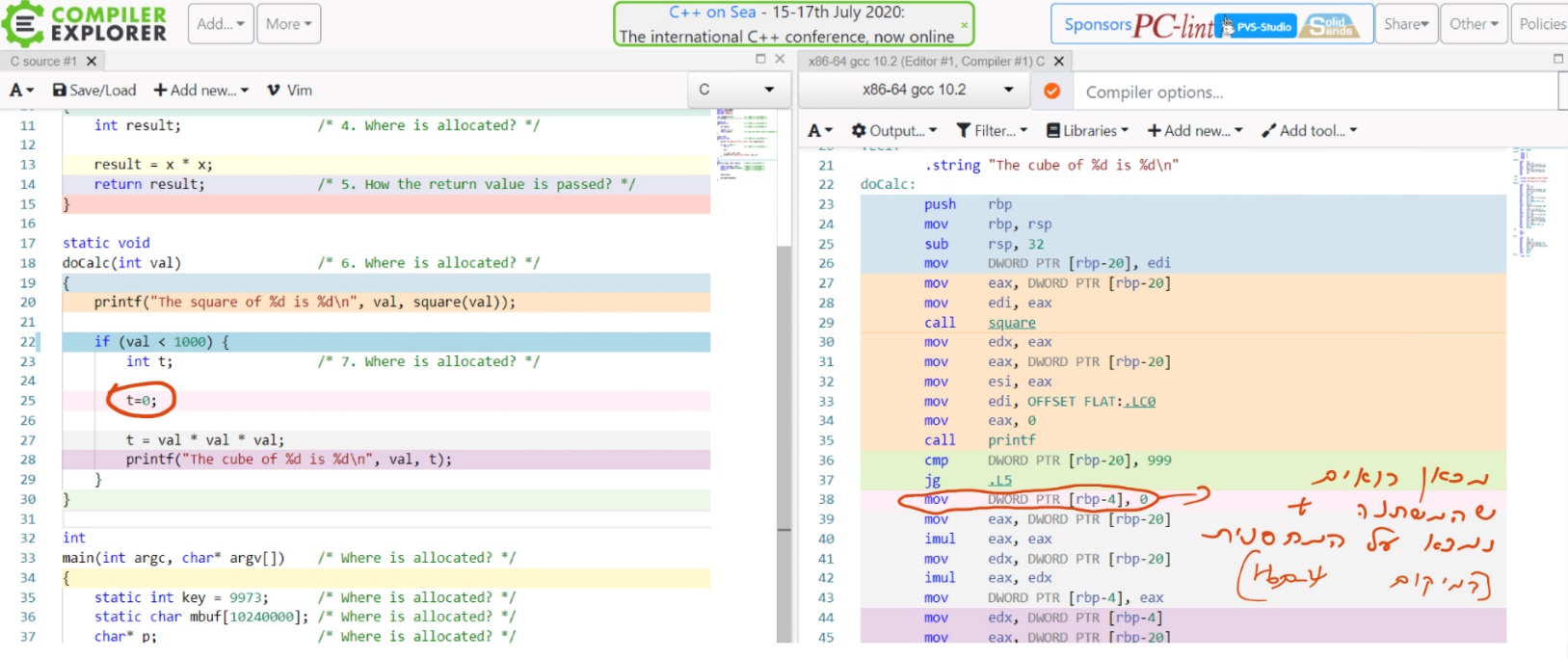
int t; */\* 7. Where is allocated? \*/*

1. המשתנה t נמצא ב stack.

בצילום הבא ניתן לראות שהשורה t=val\*val\*val מתבצעת והתוצאה נשמרת בסופו של דבר על המחסנית בהיסט של 4- מתחילת ה frame של המחסנית של הפונקציה ולכן מוכח מכאן שהמשתנה t נמצא על המחסנית.



בצילום הבא ניתן לראות את קוד האסמבלי של התכנית (מהאתר godbolt) עם הוספה של שורה 25 (t = 0), ניתן לראות מקוד האסמבלי של שורה זו שהמשתנה t נמצא על המחסנית במיקום rbp-4.



int main(int argc, char\* argv[]) */\* 8. Where is allocated? \*/*

1. הפונקציה main נמצאת ב text segment.

**static** int key = 9973; */\* 9. Where is allocated? \*/*

1. המשתנה key נמצא ב initialized data segment.

לפי מבנה הזכרון של תכניות בשפת c, משתנים סטטיים יכולים להימצא או ב uninitialized data segment או ב initialized data segment כאשר המשתנה ממוקם בחלק הראשון אם המתכנת לא איתחל אותו ובחלק השני כאשר המתכנת איתחל את המשתנה.

מאחר ו key אותחל אז הוא ממוקם ב initialized data segment.

\*מופיעים בסוף הקובץ צילומי מסך שמאשרים קביעה זו.

**static** char mbuf[10240000]; */\* 10. Where is allocated? \*/*

1. המשתנה mbuf נמצא ב uninitialized data segment (bss).

לפי מבנה הזכרון של תכניות בשפת c, משתנים סטטיים יכולים להימצא או ב uninitialized data segment או ב initialized data segment כאשר המשתנה ממוקם בחלק הראשון אם המתכנת לא איתחל אותו ובחלק השני כאשר המתכנת איתחל את המשתנה.

מאחר ו mbuf לא אותחל אז הוא ממוקם ב uninitialized data segment.

\*מופיעים בסוף הקובץ צילומי מסך שמאשרים קביעה זו.

char\* p; */\* 11. Where is allocated? \*/*

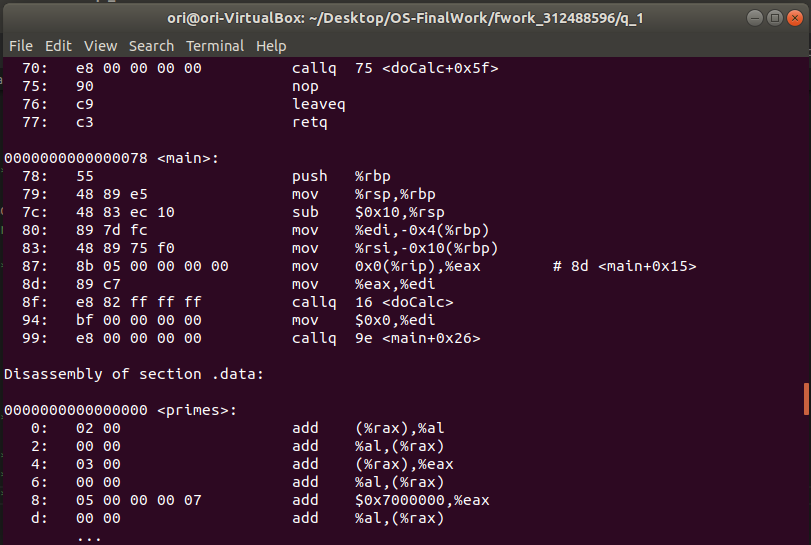
1. המשתנה p לא ממוקם בשום אזור זכרון.

תחילה נשים לב ש p הינו משתנה מקומי ששייך ל main ולכן אינו יכול להיות שייך ל initialized data segment ול uninitialized data segment (מכיוון שרק משתנים סטטיים וגלובליים ממוקמים שם) ול text segment (כי רק פקודות ממוקמות שם), בנוסף ניתן לראות ש p לא מופיע בפלטים של הפקודות שמופיעים בסוף הקובץ שמראים את מה שנמצא באזורי הזכרון הללו.

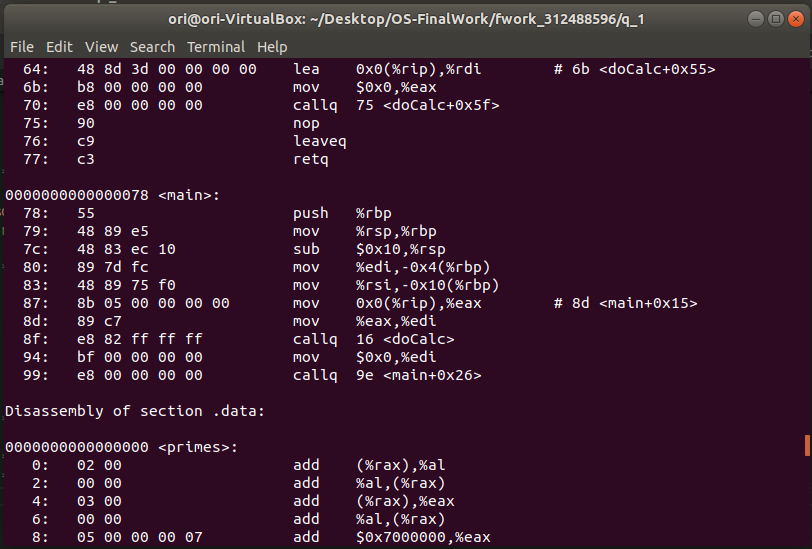
כמו כן p לא נמצא על ה heap מאחר ולא התבצעה הקצאת זכרון דינמית, לכן האפשרות היחידה שנותרה היא ש p נמצא על המחסנית.

על מנת להראות ש p לא נמצא על המחסנית הוספתי צילום מסך של קוד האסמבלי של ה main פעם אחת עם ההצהרה של המשתנה p (כמו התכנית המקורית) ופעם אחרת ללא הצהרה של המתשנה p, ניתן לראות ששניהם הקודים של האסמבלי זהים לחלוטין ולכן ניתן להסיק מכך שלמשתנה p לא מוקצה כלל זכרון.

הצילום הבא הוא הוא של קוד האסמבלי של ה main עם ההצרה על המשתנה p.



הצילום הבא הוא של קוד האסמבלי של ה main ללא ההצהרה של המשתנה p,ניתן לראות שהקודים זהים לחלוטין ומכאן ניתן להסיק שהמשתנה p לא נמצא על המחסנית.

בסופו של דבר קיבלנו שהמשתנה p לא נמצא באף אזור זכרון.

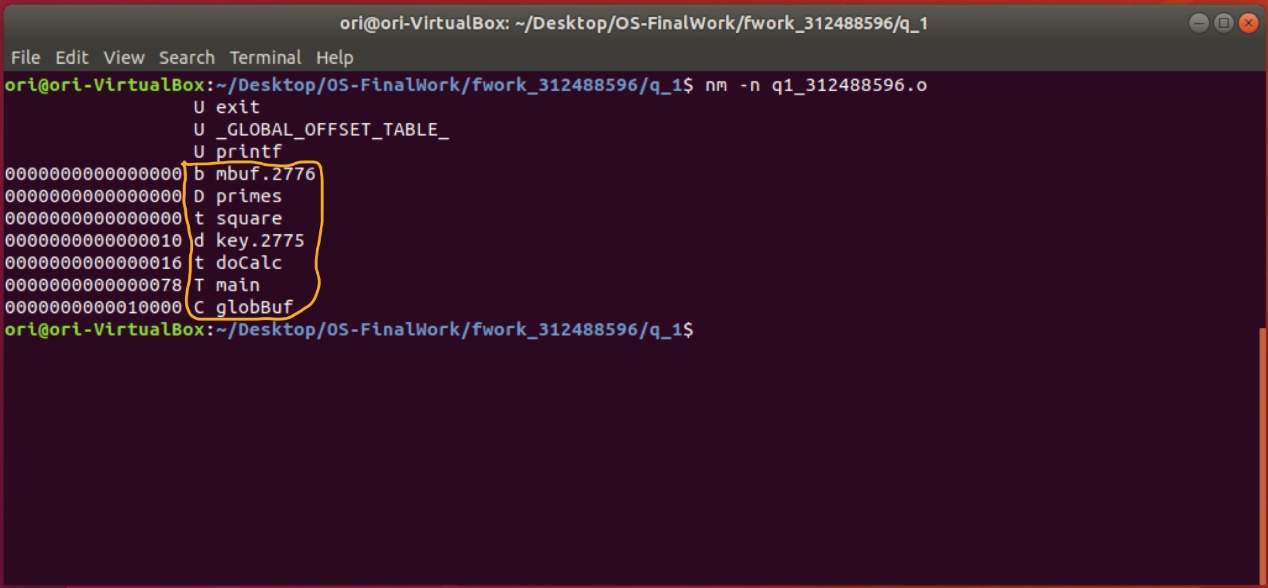
מצורפים כאן צילומי מסך של שימוש בכלים בלינוקס שמאשרים את התשובות שלי עבור שאלות 1,2,3,6,8,9,10:

בתמונה א ניתן לראות את הפלט של הפקודה nm -n על הקובץ המקומפל של התכנית, עבור כל אחד מהשמות של המשתנים/הפונקציות שמופיעות בפלט מופיעה אות באנגלית שמציינת את המקום שבו נמצא המשתנה/הפונקציה בזיכרון (על פי ה man) –

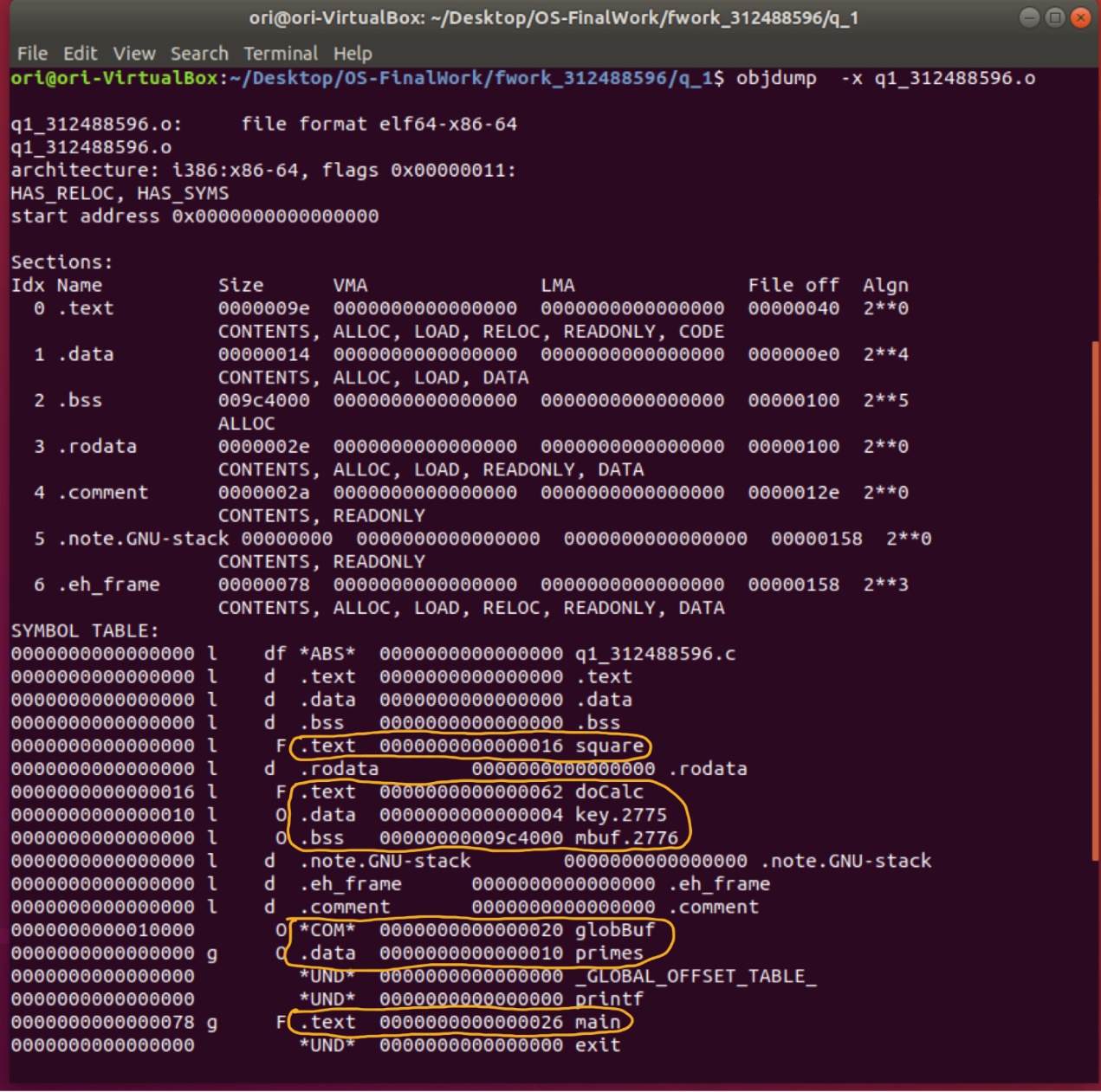
b/C – uninitialized data section

D/d – initialized data section

T/t – text section

א.

בתמונה ב ניתן לראות את הפלט של הפקודה objdump -x על הקובץ המקומפל של התוכנית, עבור כל אחד מהשמות של המשתנים/הפונקציות שמופיעות בפלט (square, doCalc, key, mbuf, globBuf, primes, main) מופיע בפירוש המקום שבו נמצא המשתנה/הפונקציה בזיכרון –

ב.