**עבודה מסכמת מערכות הפעלה:**

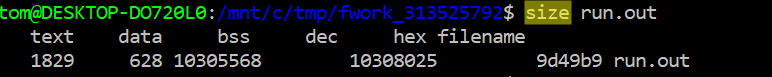
**שאלה 1:**

**סעיף 0:**

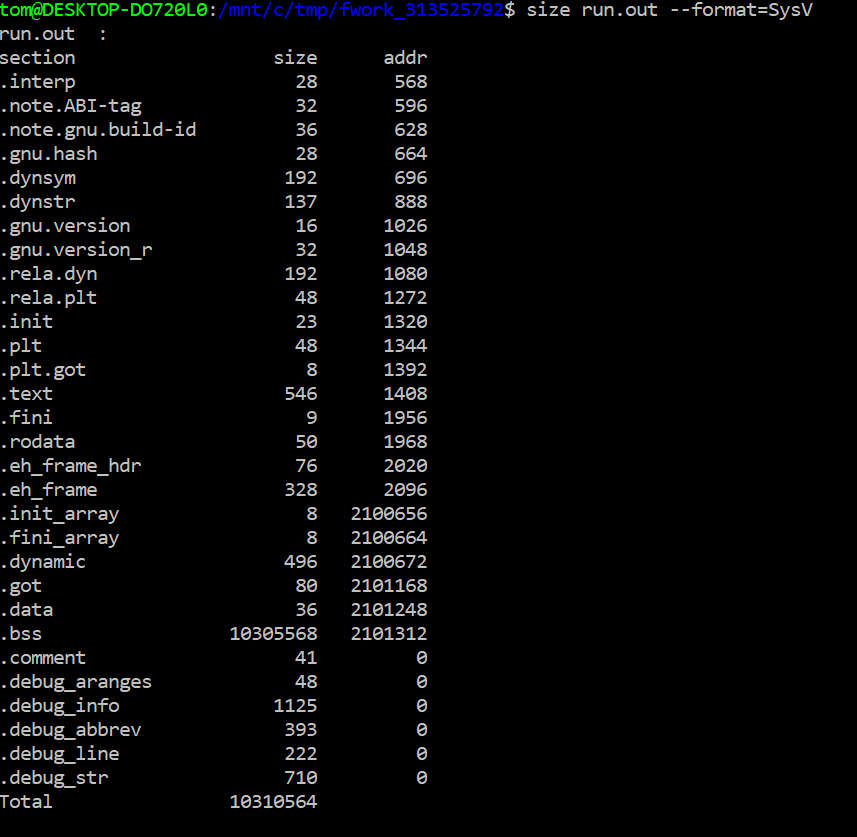
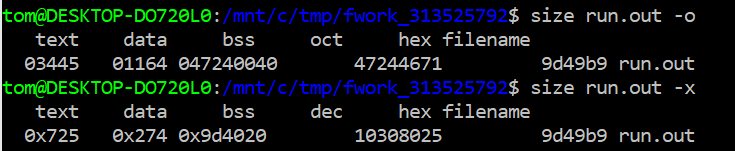
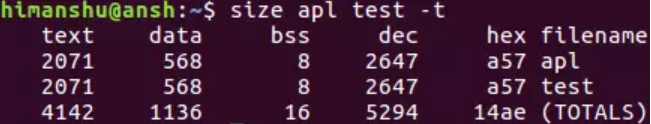
**Size:**

על מנת שנוכל לראות מה גודל כל אזור בזיכרון עבור תוכנית מסוימת נשתמש בsize-.

למשל:



לפקודה זו יש כמה אופציות שניתן להוסיף ע"י הוספת דגלים לפקודה, למשל:

* שינוי פורמט ההצגה על ידי –format=SysV  
  
* כברירת מחדל, גודל המקטעים מוצג בבסיס עשרוני.   
  עם זאת, נוכל לקבל מידע זה באופן אוקטלי או הקסדצימלי.   
  לשם כך, השתמש באפשרויות שורת הפקודה -o או -x.  
  
* אפשר לראות את גודל האזורים של כמה קבצים באופן הבא: size -t [file1] [file2]…   
  

יש עוד אופציות נחמדות אפשר פשוט להסתכל בman- או באינטרנט ולקרוא עליהם.

**Nm:**

הפקודה nm מציגה בעיקרון מידע הקשור לסמלים בקובץ אובייקט.

מה האותיות אומרות אם היא מופיעה ליד הסמל:

* A: ערך הסמל מוחלט ולא ישתנה על ידי קישור נוסף.
* b\B: הסמל נמצא בקטע הזיכרון של הנתונים הבלתי מאותחלים (המכונה BSS).
* D\d: הסמל נמצא בקטע הזיכרון של הנתונים המאותחלים.
* T\t: הסמל נמצא בקטע הזיכרון של הטקסט כלומר הקוד.
* U: הסמל לא מאותחל

\*אם זה אות קטנה זה אומר שהסמל הוא לוקאלי

לפקודה זו יש כמה אופציות שניתן להוסיף ע"י הוספת דגלים לפקודה, למשל:

* כברירת מחדל, הסמלים ממוינים לפי האלף-בית. עם זאת, אפשר למיין אותם בצורה מספרית לפי הכתובות שלהם באמצעות אפשרות שורת הפקודה -n.

יש עוד אופציות נחמדות אפשר פשוט להסתכל בman- או באינטרנט ולקרוא עליהם.

**Objdump:**

הצגת מידע על קבצי אובייקט. מציג את הדברים באופן בו אנחנו מבקשיםלפי האופציות.

אופציות:

* -d: מציג את קוד האסמבלי של קבצי אובייקט. אופציה זו מציגה רק חלקים שהם מכילים הוראות.
* -S: הצג קוד מקור שמשולב עם disassembly, במידת האפשר.

יש עוד אופציות נחמדות אפשר פשוט להסתכל בman- או באינטרנט ולקרוא עליהם.

**סעיף 1:**



**סעיף 2:**

1. שאלה: Where is allocated?  
   שורה: שורה 5

תשובה: באזור data segment - BSS, באזור זה נשמרים משתנים גלובליים וסטטיים שמאותחלים ב- 0 או שאינם מאותחלים כלל בקוד התוכנית.



1. שאלה: Where is allocated?

שורה: שורה 6

תשובה: באזור- Data data segment, באזור זה נשמרים משתנים גלובליים וסטטיים שמאותחלים בקוד התוכנית (מלבד אלו שמאותחלים ל-0) וכן מחרוזות שהוגדרו בקוד התוכנית ולא ניתנות לשינוי.



1. שאלה: Where is allocated?

שורה: שורה 9

תשובה: באזור text , אזור זה המכיל העתק של אוסף הפקודות של התוכנית.



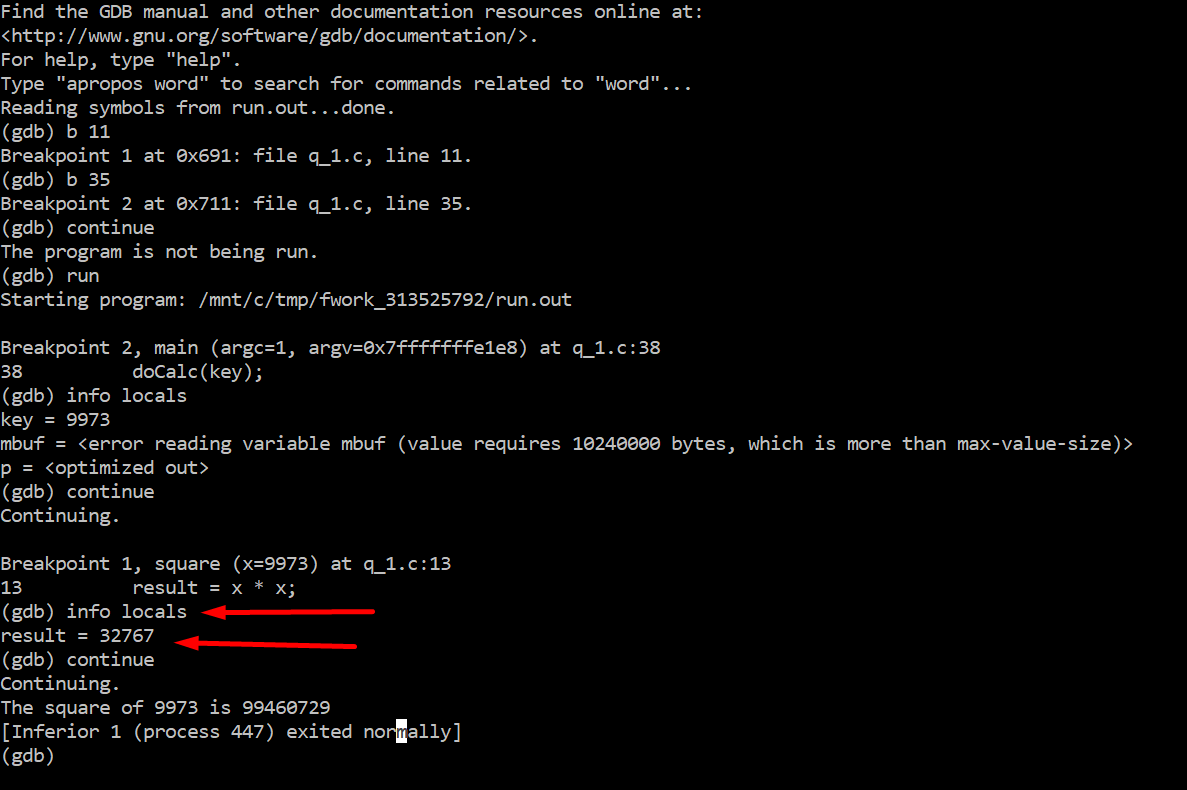
4.שאלה: Where is allocated?

שורה: שורה 11

תשובה: באזור Stack

קימפלתי באופן הבא: gcc -g q\_1.c -o run.out  
ואז הרצתי עם GDB: gdb run.out

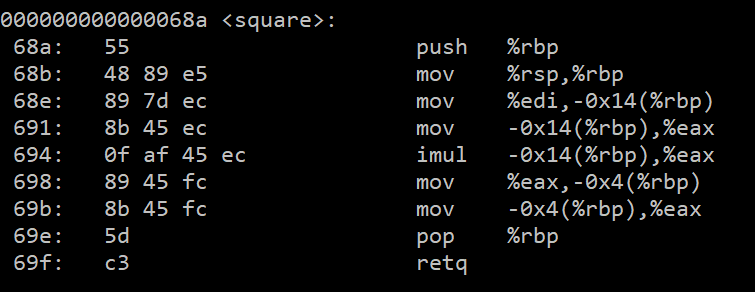
לפני שהרצתי שמתי breakpoint בשורה 11 ואחרי שהרצתי והגעתי ל-break point הפעלתי את הפקודה "info locals" שמדפיסה את כל המשתנים המקומיים.



5. שאלה: How the return value is passed?  
 שורה: שורה 14

תשובה: הערך המוחזר ע"י רגיסטר

הרצתי objdump -d run.out



6. שאלה: Where is allocated?

שורה: שורה 18

תשובה: באזור text,  הוא מקטע המכיל העתק של אוסף הפקודות של התוכנית.



7.שאלה: Where is allocated?

שורה: שורה 23

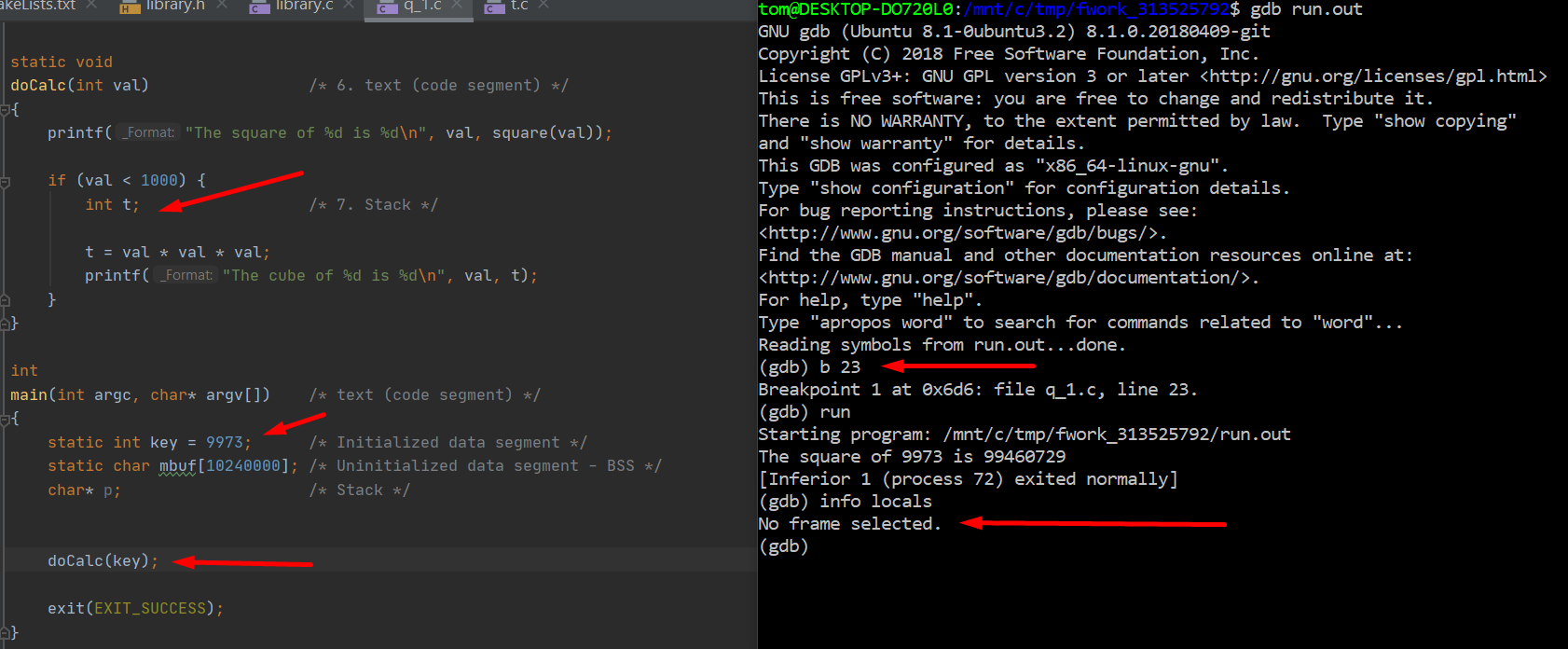
תשובה: באזור ה- Stack  
נשים לב שהמשתנה t נמצא בתוך בלוק if ולכן אם לא נכנס לתוך הבלוק הזה המשתנה t לא ייווצר על ה- Stackוכך באמת קורה מכוון שבmain- הערך אותו אנחנו שולחים לdoCalc הוא static int key = 9973 שהוא בהחלט לא קטן מ-1000 ולכן t לא נוצר ולכן אם אני יעשה את:

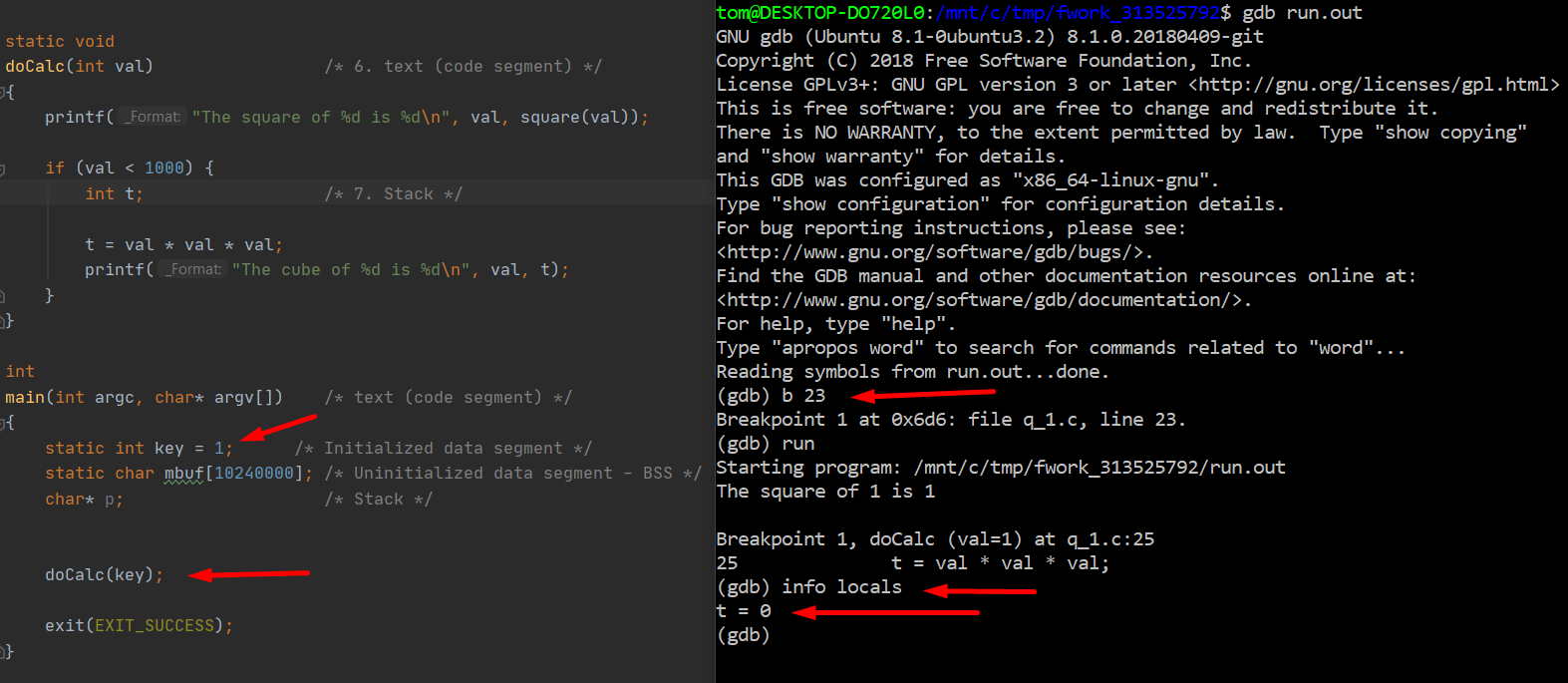
קימפול באופן הבא: gcc -g q\_1.c -o run.out  
ואז הרצה עם GDB: gdb run.out

לפני ההרצה שמתי breakpoint בשורה 23 ואחרי שהרצתי והגעתי ל-break point הפעלתי את הפקודה "info locals" שמדפיסה את כל המשתנים המקומיים אבל t לא נוצר ולכן הוא לא מודפס.

אבל אם אני ישנה את הערך של key לקטן מ-1000 אז במשתנה t כן יופיע כאשר אני יפעיל את הפקודה "info locals" .

ואכן כך:

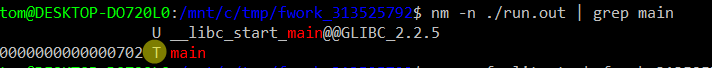




8. שאלה: Where is allocated?

שורה: שורה 31

תשובה: באזור text,אזור זה הוא מקטע המכיל העתק של אוסף הפקודות של התוכנית



9. שאלה: Where is allocated?

שורה: שורה 33

תשובה: באזור- Data data segment, באזור זה נשמרים משתנים גלובליים וסטטיים שמאותחלים בקוד התוכנית(מלבד אלו שמאותחלים ל-0) וכן מחרוזות שהוגדרו בקוד התוכנית ולא ניתנות לשינוי.



10. שאלה: Where is allocated?  
 שורה: שורה 34

תשובה: באזור data segment - BSS, באזור זה נשמרים משתנים גלובליים וסטטיים שמאותחלים ב- 0 או שאינם מאותחלים כלל בקוד התוכנית.



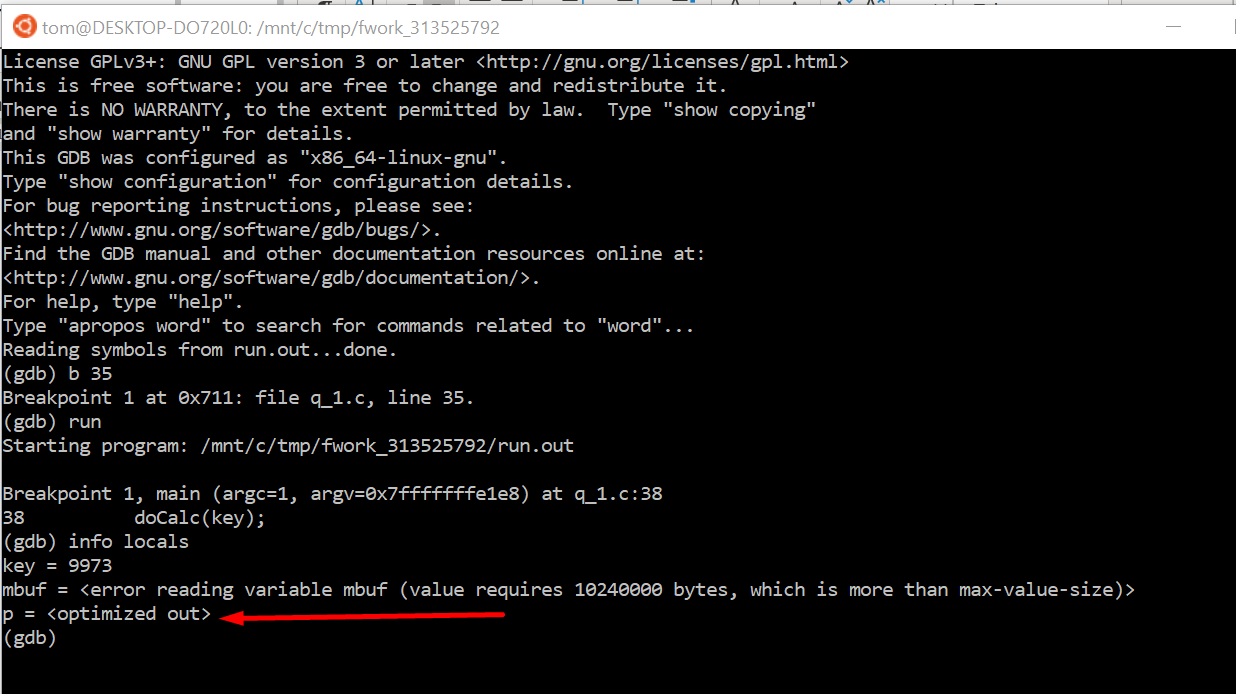
11. שאלה: Where is allocated?

שורה: שורה 35

תשובה: stack

קימפלתי באופן הבא: gcc -g q\_1.c -o run.out  
ואז הרצתי עם GDB: gdb run.out

לפני שהרצתי שמתי breakpoint בשורה 35 ואחרי שהרצתי והגעתי ל-break point הפעלתי את הפקודה "info locals" שמדפיסה את כל המשתנים המקומיים שכידוע נמצאים על ה- stack.



מקורות לשאלה:

<https://sourceware.org/gdb/current/onlinedocs/gdb/Frame-Info.html#index-info-locals-435>

<https://stackoverflow.com/questions/6261392/printing-all-global-variables-local-variables>