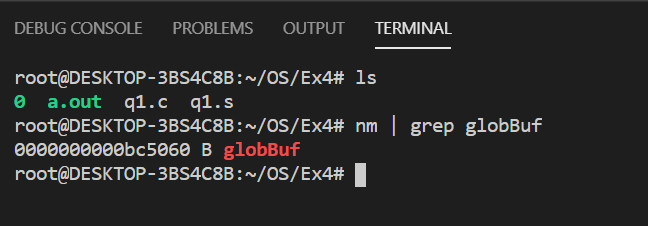
1. א)

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

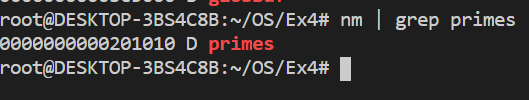
משתנים גלובלים נמצאים באזור הdata ובפרט global buf נמצא באזור של המשתנים הלא מאותחלים שם שנקרא BSS בפקודת nm רואים זאת ע"י הסימון B כקיצור לBSS .



ב)

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

משתנים גלובלים נמצאים באזור הdata ובפרט primes נמצא באזור של המשתנים המאותחלים שם בפקודת nm רואים זאת ע"י הסימון D.

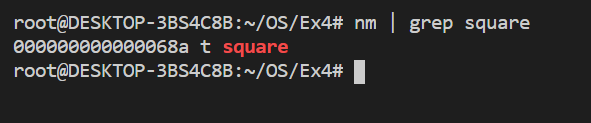


ג)

**static** void

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

הוראות הפונקציה square נמצאים באזור הtext המסומל באות t בפקודת nm .



ד)

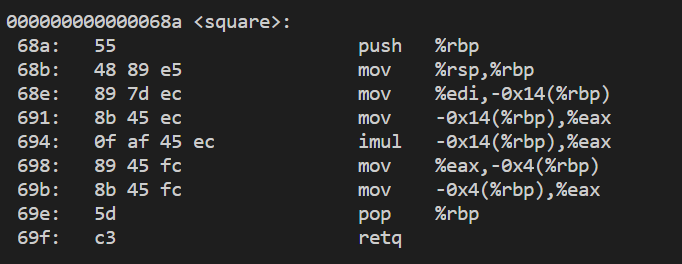
int result; */\* 4. Where is allocated? \*/*

המשתנה result נמצא על המחסנית במקטע שנוצר בשביל הפונקצייה square

קודם כל על דרך השלילה לא התקבל כל קלט עי פקודת nm



מה שאומר שresult לא נמצא בdata segement ולא בtext/code segment וברור שהוא לא נמצא בHeap כי לא השתמשנו בmalloc



לעיל מקטע האסמבלי השייך לפונקציית square

ניתן לראות שאחרי פקודות הפתיחה שמסמנות frame חדש בשביל הפונקצייה square

אנו מקצים 5 מקומות בstack (קפיצות של 4) מעתיקים האת התוכן הארגמנט שנשלח לפונקצייה (נמצא ברגיסטר edi ) למקום ה5 מההתחלה(%rbp )

(נקודה חשובה שהבחנתי שאנו לא משתמשים בsub כי אין צורך לסמן את הסיום החדש כי במילא הפונקצייה לא קוראת לפונקצייה נוספת והיא האחרונה בשרשרת ולכן בסיומה נחזור למסגרת הקודמת)

מעתיקים את הערך משם (כרגע זה הארגמנט שנשלח לפונקצייה) למקום בו אנו שומרים את הערך המוחזר מהפונקצייה (רגיסטר eax ) – לפי דעתי משמש כרגע בתור אוגר רגיל

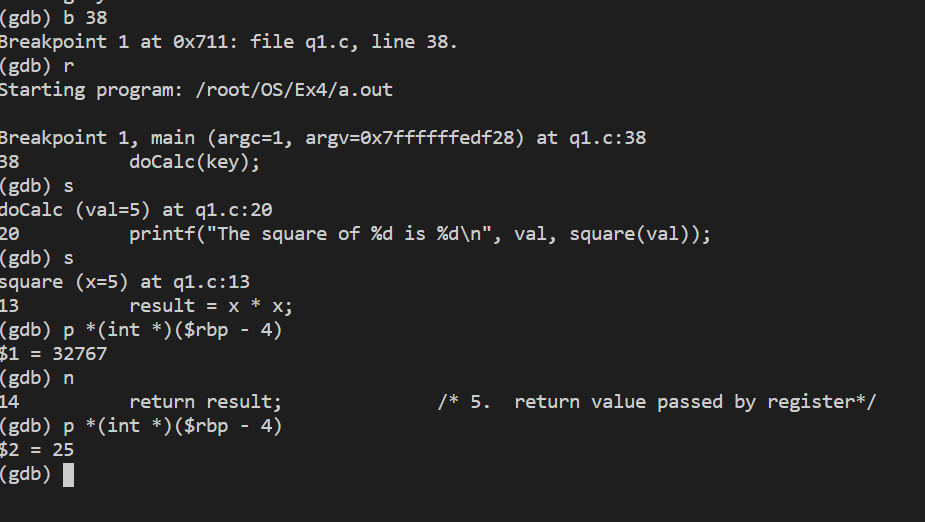
כעת נכפיל מה שנמצא באוגר eax בתוכן של מה שנמצא במקום החמישי (הארגמנט שנשלח לפונקצייה) – כלומר מבצע x\*x

**לאחר מכן הוא מעתיק את התוצאה גם למקום הראשון שהקצנו** (%rbp -4 )

מעתיק גם למקום של הערך המוחזר ( שקול לפקודה return)

ההעתקה למקום הראשון שהקצנו את התוצאה של ההכפלה של הארגמנט שנשלח לפונקציה – זה בדיוק מה שהמשתנה result שומר ולכן result ממוקם על המחסנית בframe של square

נריץ gdb ונשים לב שכאשר נשמר ערך לתוך result המקום הזה במחסנית משנה את ערכו לערך הנכון של 5\*5 (שיניתי את ערך המשתנה המקורי) = 25



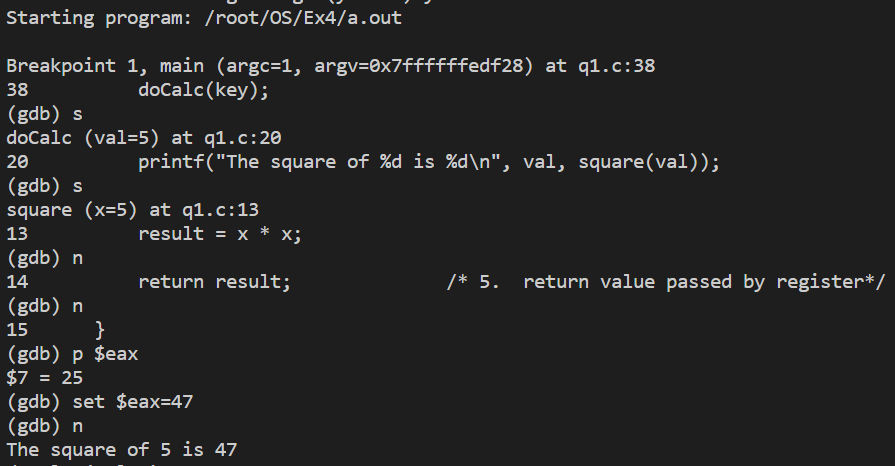
ה)

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

הערך מוחזר ע"י רגיסטר מתאים והתוכנית הקוראת לפונקציה יודעת לקרוא מאותו רגיסטר את הערך המוחזר

נסתכל על הניתוח שביצעתי בסעיף קודם ונבחין שערך ההחזרה מעובר עי הרגיסטר eax

נוסיף הוכחה עי הgdb כאשר נשנה את התוכן של האוגר אחרי שנגמרה הפונקצייה ונבחין שהפונקצייה הקוראת קיבלה את הערך המשונה

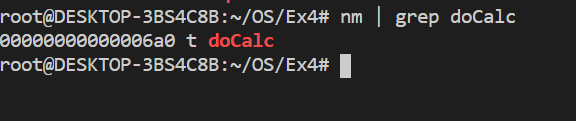


ו)

**static** void

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

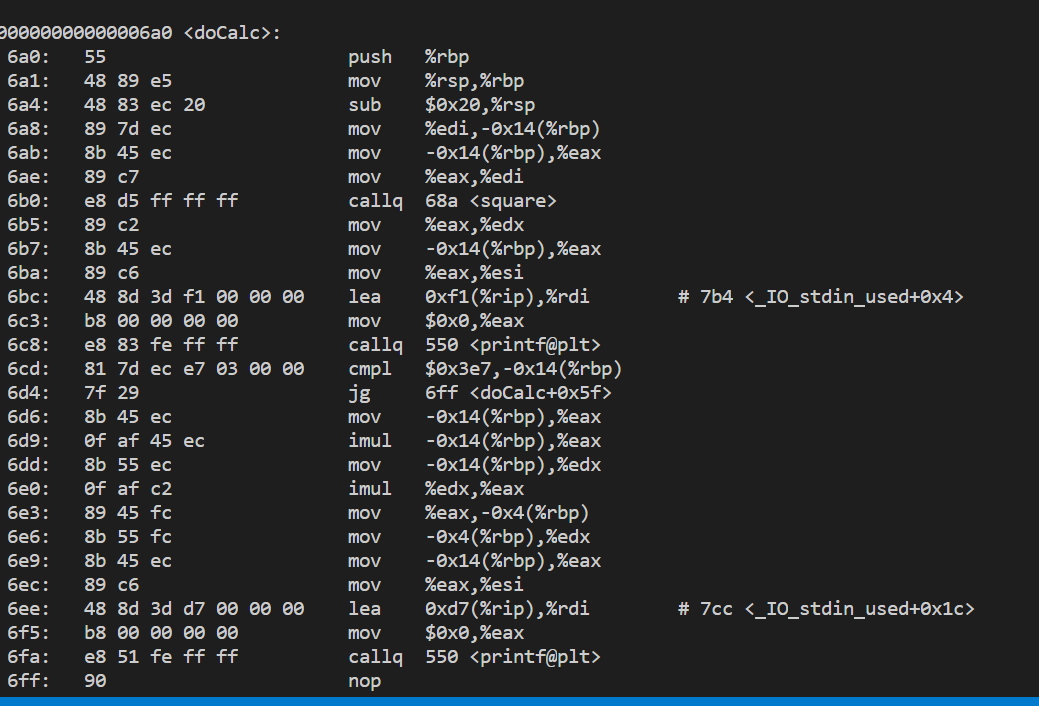
הוראות הפונקציהdoCalc נמצאים באזור הtext המסומל באות t בפקודת nm .



ז)

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

המשתנה t נמצא על המחסנית במקטע שנוצר בשביל הפונקצייה doCalc



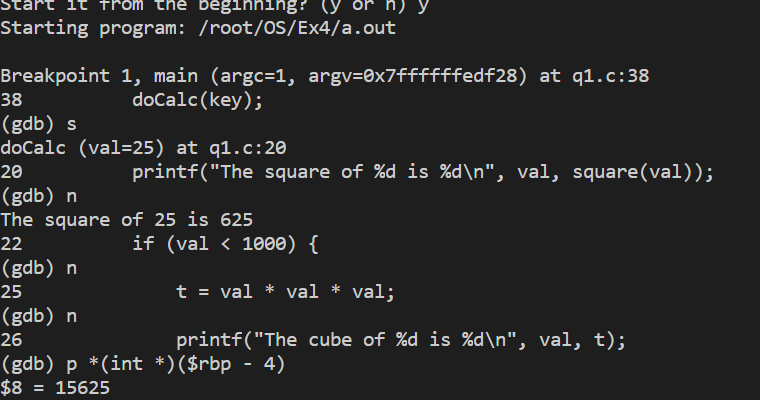
להלן קטע עיקר קטע האסמבלי של הפונקצייה doCalc נוכל להתרכז בשלב של הcmpl שם בעצם מתבצי הif

לאחריו מתבצעות מספר הכפלות באמצעות אוגרים ולבסוף התוצאה שמתקבלת באוגר eax מעוברת למקום הראשון שהוקצה במחסנית למסגרת של הפונקצייה doCalc כלומר

ל (%rbp -4 ) ולכן המשתנה t שבעצם מאחסן את התוצאה של העלאת הארגמנט בשלישית נמצא על המחסנית במקום (%rbp -4 ) ש%rbp מציין את התחלת המסגרת של פונקציית doCalc

נוודא זאת עי שינוי התוכנית ונקרא לפונקצייה עם ארגמנט קטן מ1000

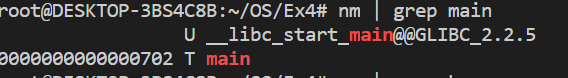
לדוגמא 25 .. ונוודא שהמקום במחסנית מכיל את הערך המדובר באמצעות gdb והתוצאה לפניכם



ח)

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

ההוראות של הפונקצייה main נמצאות באזור הtext והמשתנים נמצאים בקטע המחסנית של הפונקצייה main



ט)

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

משתנים סטטים נמצאים באזור הdata

Key מאותחל ולכן נמצא בתת אזור של data המאחסן משתנים מאותחלים המסומל בפקודת nm באות d



י)

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

משתנים סטטים נמצאים באזור הdata

mbuf לא מאותחל ולכן נמצא בתת אזור של המשתנים הלא מאותחלים bss- המסומל בפקודת nm באות b



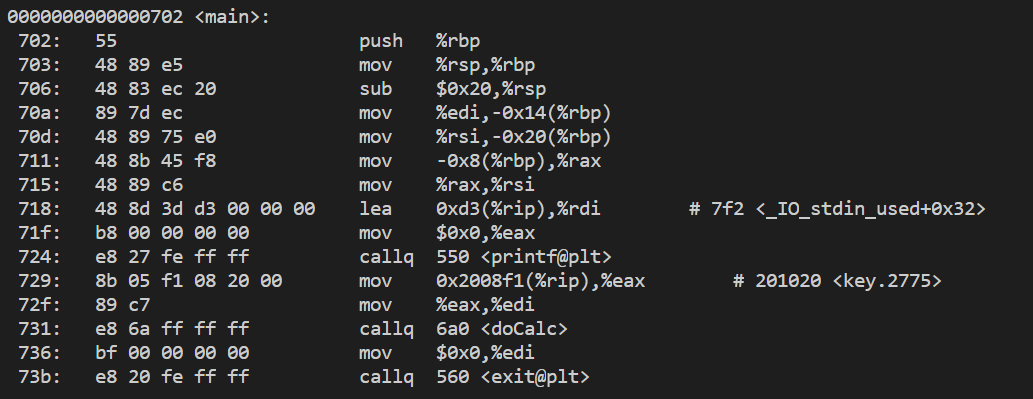
יא)

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

לכאורה אמור להיות מאותחל על stack במקטע ששייך לmain אך לאחר כל מיני חקירות

ראיתי שאם מוחקים אותו, הקצאת הזיכרון נשארת אותו דבר מצד שני אם אני משתמש בו איכשהוא בהדפסה או משהו אחר הקצאת הזיכרון על המחסנית גדלה

לכן הסקתי שהדבר נובע מאופטימיזציה של הקומפיילר שרואה שאין בו שימוש ולכן מראש לא מקצה אותו בכלל

עם שימוש במשתנה 

בלי שימוש

