שאלה מספר 1-

לצורך מענה על שאלה זו השתמשתי בשני כילים : nm, objdump (תמונות של הרצת הפקודות הללו מצורפות מטה).

כפי שניתן לראות בטרמינל, עבור הסמל globBuf קיבלתי את האות 'B' . uninitialized data section . והמשמעות היא שהסמל

```
int primes[] = { 2, 3, 5, 7 }; // 2. Initialized data section.2
00000000000000000 D primes
```

עבור הסמל **primes** קיבלתי את האות '**D'** והמשמעות היא שהסמל נמצא בinitialized data section.

```
square(int x) //3. allocated in frame for square().3

0000000000001169 t square
```

עבור הסמל square קיבלתי את האות 't' והמשמעות היא שהסמל נמצא ב-Text section.

```
int result; // 4. allocated in the Stack of square() function .4

4: 55 push %rbp
5: 48 89 e5 mov %rsp,%rbp
```

בשאלה זו השתמשתי בפקודה **- objdump -d**. כפי שניתן לראות בטרמינל - בשאלה זו השתמשתי בפקודה int result הוצהר- מה שקרה מאחורי הקלעים square() של הפונקציה (square (ע"י הפקודה result) ולכן ניתן להסיק שהמשתנה result ממוקם על stack.

רesult בשאלה זו השתמשתי בפקודה **objdump -d**. בסעיף 4 דחפנו את push בשאלה זו השתמשתי בפקודה push ע"י רגיסטר presult ובסוף כפי שניתן לראות בטרמינל - מתבצע pop למשתנה result ולכן ניתן להסיק כי הוא מועבר ע"י רגיסטר.

doCalc(int val) // 6. allocated in frame for doCalc() .6

0000000000001182 t doCalc

עבור הסמל doCalc קיבלתי את האות 't' והמשמעות היא שהסמל נמצא ב-Text section.

כפי שניתן לראות בטרמינל, jg זה התנאי if שלפני הצהרת המשתנה t. כשר נכנסנו לתנאי (if), ניתן לראות כי מתבצע mov למשתנה t ולכן אפשר להסיק כי המשתנה t נמצא על הstack.

main(int argc, char* argv[]) // 8. allocated in frame for main() .8

00000000000011e7 T main

עבור הסמל **main** קיבלתי את האות **'T'** והמשמעות היא שהסמל נמצא ב-Text section.

static int key = 9973;// 9. Initialized data section .9

00000000000004020 d key.2841

עבור הסמל **key** קיבלתי את האות **'d'** והמשמעות היא שהסמל נמצא ב- **initialized data section**

static char mbuf[10240000]; // 10. Uninitialized data section .10

0000000000004060 b mbuf.2842

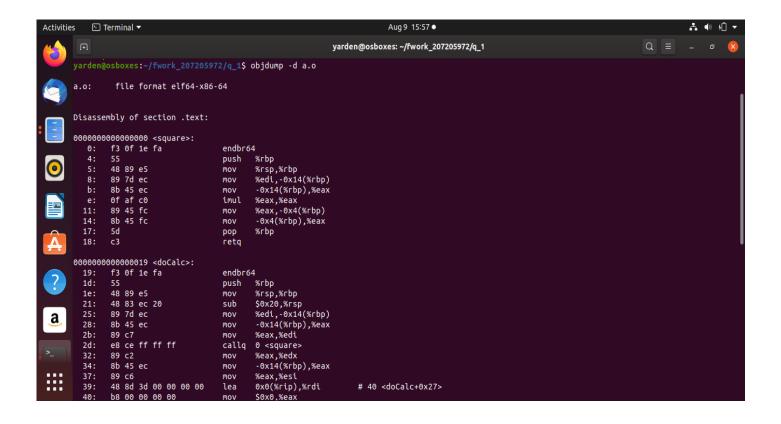
עבור הסמל **mbuf** קיבלתי את האות **'b'** והמשמעות היא שהסמל נמצא בuninitialized data.

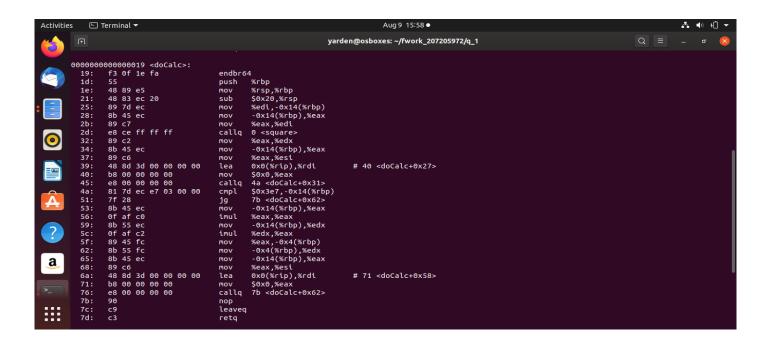
```
char* p; // 11. allocated in the Stack .11

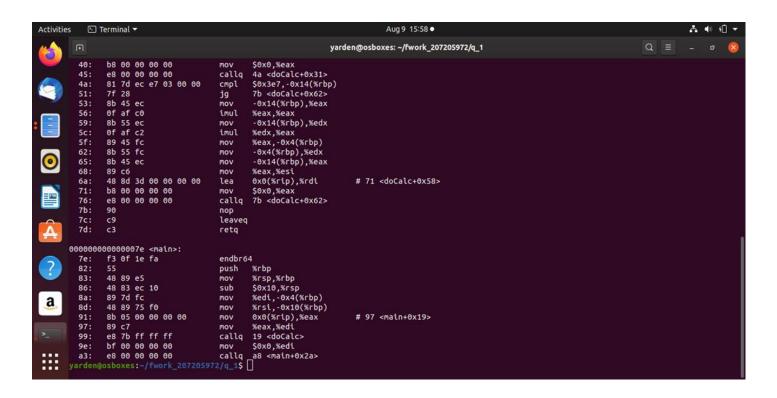
82: 55
83: 48 89 e5
86: 48 83 ec 10
80: 50x10, xrsp
80: 89 70 fc
80: 48 89 75 f0
```

כפי שניתן לראות בטרמינל - כאשר המשתנה p הוצהר , מה שקרה מאחורי p הקלעים זה ש- p הוכנס לתוך הtack (ע"י הפקודה p ולכן ניתן להסיק שהמשתנה p ממוקם על stack.

<u>: objdump –d הרצת הפקודה</u>







<u>הרצת הפקודה nm-</u>

