# שאלה מספר 1-

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

#### 1. where is allocated?

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

int primes[] = { 2, 3, 5, 7 }; // 2. Initialized data section.2
000000000000000000 D primes

#### 2. Where is allocated?

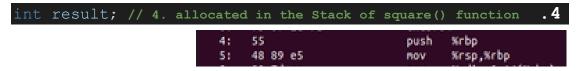
עבור הסמל **primes** קיבלתי את האות 'D' והמשמעות היא שהסמל נמצא בinitialized data section.

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

000000000001169 t square

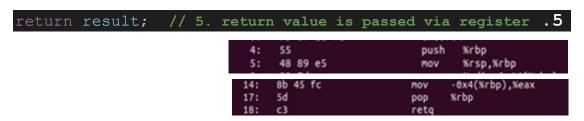
### 3. Where is allocated?

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



#### 4. Where is allocated?

בשאלה זו השתמשתי בפקודה - objdump -d. כפי שניתן לראות בטרמינל - בשאלה זו השתמשתי בפקודה - מה square של () square כאשר square (ע"י הפקודה result ולכן ניתן להסיק שהמשתנה result ממוקם על stack.



#### 5. How the return value is passed?

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

#### 6. where is allocated?

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

#### 7. Where is allocated?

כפי שניתן לראות בטרמינל, jg זה התנאי if שלפני הצהרת המשתנה t. כאשר 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
```

## 8. Where is allocated?

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

#### 9. Where is allocated?

עבור הסמל **key** קיבלתי את האות **'d'** והמשמעות היא שהסמל נמצא בinitialized data section.

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

000000000000000000 b mbuf.2842
```

## 10. Where is allocated?

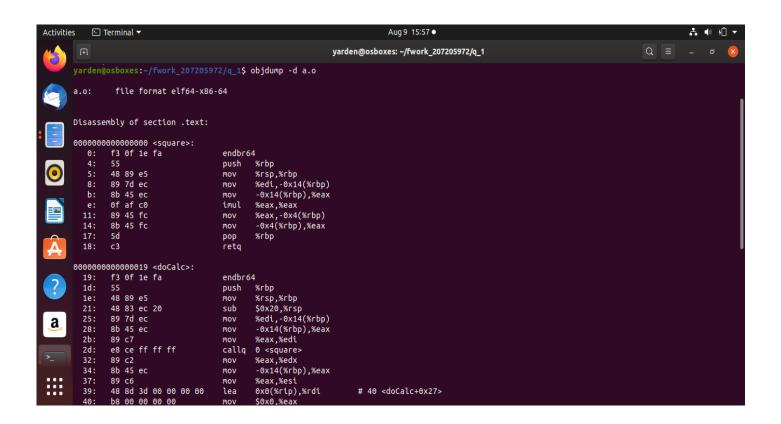
עבור הסמל **mbuf** קיבלתי את האות **'b'** והמשמעות היא שהסמל נמצא בuninitialized data.

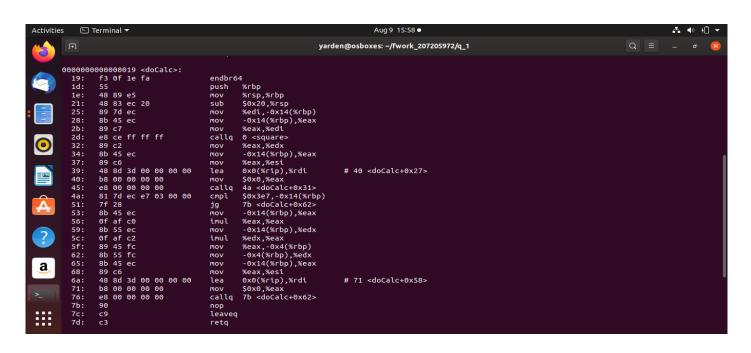
```
11. allocated in the Stack .11
char* p;
                               83:
                                    48 89 e5
                                                                 %rsp,%rbp
                                                           nov
                               86:
                                    48 83 ec 10
                                                           sub
                                                                  $0x10,%rsp
                                                                 %edi,-0x4(%rbp)
                                    89 7d fc
                               8a:
                                                           mov
                                    48 89 75 f0
                                                                 %rst,-0x10(%rbp)
                               8d:
                                                           nov
```

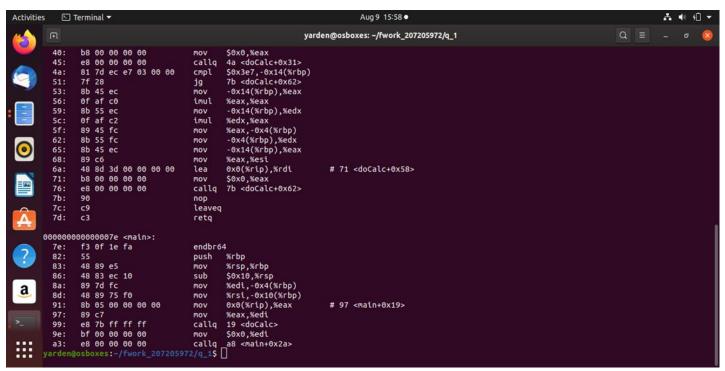
#### 11. Where is allocated?

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

### : objdump –d הרצת הפקודה







#### <u>הרצת הפקודה nm-</u>

