### תז: 1. 322315318

2. 208456285

### ‘

### הטכניון – מכון טכנולוגי לישראל

**ארגון ותכנות המחשב**

תרגיל 2 - חלק יבש

המתרגל האחראי על התרגיל: תומר כץ.

שאלותיכם במייל בעניינים מנהלתיים בלבד, יופנו רק אליו.

**כתבו בתיבת subject: יבש 2 את"ם.**

שאלות בעל-פה ייענו על ידי כל מתרגל.

הוראות הגשה:

* לכל שאלה יש לרשום את התשובה במקום המיועד לכך.
* יש לענות על גבי טופס התרגיל ולהגיש אותו באתר הקורס כקובץ PDF.
* על כל יום איחור או חלק ממנו, שאינו בתיאום עם המתרגל האחראי על התרגיל, יורדו 5 נקודות.
* הגשות באיחור יש לשלוח למייל של אחראי התרגיל בצירוף פרטים מלאים של המגישים (שם+ת.ז).
* שאלות הנוגעות לתרגיל יש לשאול דרך הפיאצה בלבד.
* ההגשה בזוגות.

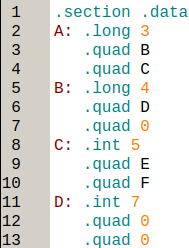
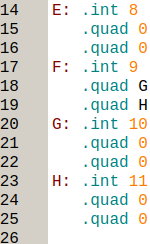
### 

# שאלה 1 (45 נק') – שגרות:

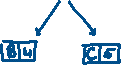
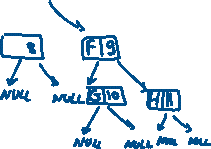


ג'וני סטודנט אחראי כל בוקר בשש ו30 כותב קוד אסמבלי. לפניכם מקטע הנתונים שג'וני כתב:



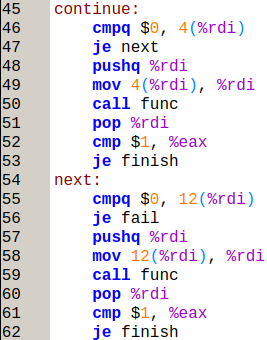
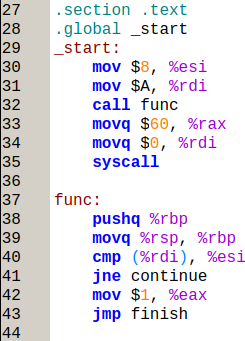
 

1. ציירו את הגרף המתקבל מפירוש מקטע הנתונים (מומלץ להסתכל בתרגול 3 תרגיל 1 ולהיזכר שם על אופן פירוש הזיכרון כרשימה מקושרת). בכל צומת בגרף ציינו את התווית המתאימה לו בלבד (אין צורך לציין ערכים נוספים) (3 נקודות)

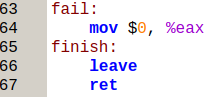


ג'וני לא מפחד משגרה שוחקת ולכן כותב את השגרה func וקוד שמתשמש בה:









1. נתון שבתחילת התוכנית ערך של rsp הוא x.כאשר x הוא מספר בקסדצימלי. מה הוא הערך המקסימלי ומה הערך המינימלי שrsp יכיל לאורך ריצת התוכנית? תנו נוסחא שהמספרים בה הם בבסיס הקסדצימלי (בטאו את התשובה בהאמצעות x). (5 נקודות)

**המחסנית גדלה כלפי מטה ולכן הערך המקסימלי של rsp הינו x**

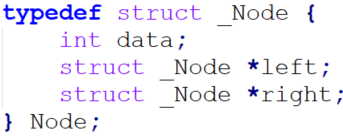
**והערך המינימלי הוא x – 0x40**

1. רשמו מה יהיה פלט הפונקציה עבור קטע הקוד הנוכחי (7 נקודות)

**הפלט יהיה 1**

1. המירו את הפונקצייה לשפת c על ידי כך שתשלימו את המקומות החסרים בקוד. העיזרו בהגדרת הstruct שנתונה לכם (10 נקודות):

הstruct הנתון:



**הערה1:** שני הפרמטרים צריכים להיות תואמים לשני הפרמטרים של פונקצית האסמבלי גם מבחינת תפקיד וגם מבחינת סדר. כלומר, root צריך להתאים בתפקידו לפרמטר הראשון שמועבר לפוקנציה בשפת אסמבלי גם מבחינת הקונבנצייה שלמדנו.



**הערה2:** אורך הקו לא מלמד על אורך האיבר שצריך להשלים. **מותר להשלים יותר ממילה אחת בכל קו אך לא יותר מפקודה אחת!**

**\_\_\_\_\_bool\int\_\_\_\_** func (Node\* root, **\_\_\_\_int\_\_\_\_\_** x){

If (root->data == **\_\_\_\_\_\_\_\_\_\_x\_\_\_\_\_\_\_\_\_\_**)



**\_\_\_\_\_\_\_\_return true\_\_\_\_\_\_\_\_**;



if (root.left != null)

if (**\_\_\_\_\_\_func(root.left,x)\_\_\_\_\_**)

return **\_\_\_\_\_true\_\_\_\_\_\_\_\_\_\_**;

if (root.right != null)

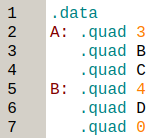
return **\_\_\_ func(root.right,x) \_\_\_\_**;

return 0;

}

הערה: בסעיפים הבאים יש כל מיני שינויים בקוד. כל שינוי מתקיים רק בסעיף בו מופיע. זאת אומרת הסעיפים לא תלויים אחד בשני.

1. מוני חבר של ג'וני הוא לא כמו ג'וני. הוא אוהב לעשות שינויים רבים בקוד. הוא מחליט לקחת את המקטע הנתונים של ג'וני ולשנות בכל struct את הint בquad. כלומר מקטע הנתונים ישתנה כך:



ובאופן דומה כל שאר האותיות יחליפו את הנתון הראשון ב-quad במקום ה-int.

רשמו את השינויים שצריכים להיות בקוד על מנת שיעבוד בצורה תקינה עם מקטע הנתונים החדש (5 נקודות)

**Lines 30 & 40: change %esi to: %rsi**

**Lines 46 & 49: change 4(%rdi) to: 8(%rdi)**

**Lines 55 & 58: change 12(%rdi) to: 16(%rdi)**

1. ג'וני מתחיל להתעייף מהשגרה ומחליט לקום ולשנות את מבנה הנתונים באופן הבא:



מה יהיה פלט התוכנית? יש לסמן תשובה מבין התשובות הבאות ולנמק בקצרה: (5 נקודות)

* התוכנית תסתיים ופלט הפונקציה יהיה 1
* התוכנית תסתיים ופלט הפונקציה יהיה b
* התוכנית תכנס ללולאה אנסופית
* התוכנית תקרוס במהלך ריצה
* התוכנית כלל לא תבנה

נימוק: **\_\_\_ נוצר מעגל ב"עץ"( (A->…->D->A\_ואנו נרוץ על המעגל הזה עד שהמקום על המחסנית ייגמר והתוכנית תקרוס**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. פתאום ג'וני כמו מוני! מחליט לבצע שינויים נוספים ולא שגרתיים בקוד מול כל שינוי שג'וני מציע עליכם לכתוב האם נכונות השגרה תיפגע (האם יש קלט עבורו השגרה לאחר השינוי שונה מהשגרה לפני השינוי). הסבירו **בקצרה** את תשובתכם! (10 נקודות)

* מחיקת הפקודת push וpop שבשורות 60 ו 57..
* מחיקת הפקודה pop בשורה 60



* מחיקת push וpop שבשורות 48 51
* הוספת פקודה push %rdi אחרי continue בשורה 45
* הוספת הפוקדה push %rdi אחרי continue בשורה 45, שינוי פוקדת הpop שבשורה 51 לפקודה: mov (%rsp), %rdi ומחיקת הפקודת push וpop בשורות 60 57.

נימוק:



**1. לא יישתנה כלום, מכיוון שזה חיפוש PREORDER ואין צורך לשמור את האב**

**2.הנכונות לא תיפגע מכיוון שבסוף הקוד ה LEAVE "יציל" אותנו ויעביר את RSP להיות RBP**

**3. לא נצליח לעשות סיור PREORDER מכיוון שלא נדע איך לחזור לאב לאחר סיור בבן השמאלי**

**4. כמו ב 2**

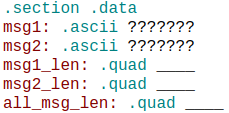
**5. כמו ב 2**

# שאלה 2 (30 נק') – קריאות מערכת:

ג'ואי מרגיש מתוסכל מכך שחבריו חושבים שהוא פחות חכם מהם. לכן, הוא מחליט להרשים אותם בעזרת כתיבת קוד אסמבלי.



1. לפניכם מקטע הנתונים שג'ואי כתב מבלי ערכי הנתונים עצמם:



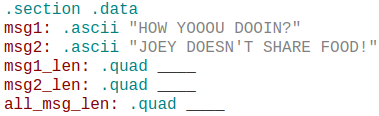
ג'ואי לא יודע עדיין אילו מחרוזות הוא יכתוב. עליכם להשלים את המקומות הריקים שקשורים לאורכי המחרוזות כך שמשתנה msg1\_len יהיה האורך של msg1, בmsg2\_len יהיה האורך של msg2 ובמשתנה all\_msg\_len יהיה שווה לסכום אורכי המחרוזות msg1 וmsg2. שימו לב עליכם לעשות זאת בצורה כזו שהאורכים יהיו נכונים בעת ריצת התוכנית ללא קשר לאיזה מחרוזות ג'ואי ישים בmsg1 ובmsg2. (3 נקודות)

a. **msg2 – msg1**

b. **msg1\_len – msg2**

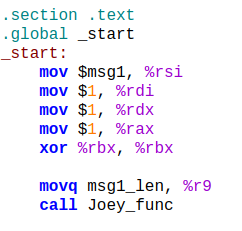
c. **msg1\_len – msg1**

1. כעת נתון מקטע הנתונים שכולל את המחרוזות:



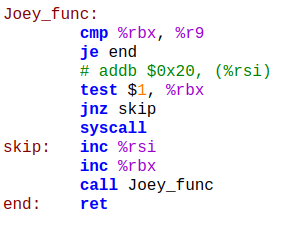


לפניכם נתונה התוכנית שג'ואי כתב:



ומוצגת כאן גם הפונקציה שכתב:





מה יודפס בסיום ריצת הקוד? (שימו לב השורה השלישית בפונקציה נמצאת בהערה ולא רלוונטית לסעיף). (5 נקודות)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_HWYOUDON\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. כעת מורידים את הסולמית שנמצאת בפונקציה (וכעת הפקודה חלק מהקוד) בנוסף מחליפים את השורה movq msg1\_len, %r9 בשורה: movq all\_msg\_len, %r9.

**הערה: שינויים אלו ילוו אותנו גם בסעיפים הבאים (בסעיפים ד - ו השינויים בסעיף ג עדיין תקפים).**

מה יודפס כעת בסיום ריצת הקוד? (5 נקודות)

**hwyoudonje@osG@hr@od**

1. בזמן שג'ואי אכל בסלון סנדוויץ, חיית המחמד שלו (אפרוח) טיילה על המקלדת והוסיפה את הפקודה:

inc %r9. הפקודה נוספה שורה לפני הקריאה לפונקציה של ג'ואי בתוכנית הראשית.

מה יהיה פלט התוכנית כעת? (2 נקודות)

**hwyoudonje@osG@hr@od0**

1. חברה טובה של ג'ואי פיבי אמרה לו ששימוש ברגיסטר r9 מביא מזל רע. ג'ואי נלחץ נורא והחליט שיש לבצע שינוי בקוד מבלי לשנות את תוצאות הפעולה של הפונקציה (כלומר הפלט צריך להיות זהה). כיוון ולא ידע איך לשנות את הקוד הוא החליט לבקש את עזרת חבריו.

בסעיף הזה יופיעו העצות של כל החברים. עליכם לרשום ליד כל עצה האם היא לדעתכם תעזור לג'ואי. נמקו **בקצרה(!)** (10 נקודות)

* + צ'נדלר מציע להחליף את השימוש בr9 בשימוש בrcx.

**לא תעזור מכיוון ש SYSCALL דורסת את RCX(שמה בו את RIP) ובכך נאבד את כמות הבתים שאנחנו רוצים לעבור עליהם**

* + מוניקה מציעה להחליף את השימוש בr9 בשימוש בr11.
* **לא תעזור מכיוון ש SYSCALL דורסת את 11R(שמה בו את RFLAGS) ובכך נאבד את כמות הבתים שאנחנו רוצים לעבור עליהם**
  + פיבי מציעה להחליף את השימוש בr9 בשימוש בrdi.

**לא יעזור מכיוון ש SYSCALL מצפה לקבל 1 ב RDI עבור הדפסה למסך, דבר שלא יקרה אם RDI יקבל את הערך במקום 9R**

* + רייצ'ל מציעה להחליף את השימוש בr9 בשימוש בr12.

**אכן יעזור, זהו רגיסטר פנוי ולא נעשה בו שימוש בשום מקום אחר**

* + רוס מציע להחליף את השימוש בr9 בשימוש בrbp.

**הקוד יעבוד, אך עלול להשפיע על המשך פעילות תקינה בהמשך**

1. חבריו של ג'ואי מסבירים לו שהשימוש שלו ברקורסיה מיותר ובזבזני והוא יכול את אותו קוד בדיוק לכתוב בלולאות. ג'ואי מחליט לבצע את השינויים הבאים:

בתוכנית הראשית בשורה שלפני ביצוע הפקודה call ג'ואי מוסיף את הפקודה:

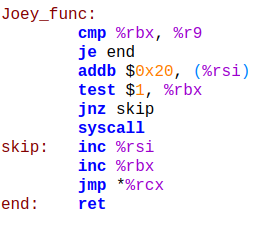
mov $Joey\_func, %rcx

ובתוך הפונקציה ג'ואי מוחק את השורה בה יש שימוש בפקודה call והחליף אותה בפקודה:

jmp \*%rcx

שימו לב שהתווית end נמצאת אחרי פקודה זו.

לצורך הבהרה הפונקציה נראת כך כעת:



כיצד שינוי זה ישפיע על אופן ריצת הפונקציה. מה יודפס אם נריץ את הפונקציה? (5 נקודות)

**תודפס האות h והפונקציה תיכנס ללואה אינסופית**

# שאלה 3 (25 נק') – רמות הרשאה ואוגר הדגלים:

1. הפקודה pushfq דוחפת את הערך של אוגר הדגלים למחסנית. והפקודה popfq מוציאה את אוגר הדגלים מהמחסנית. הסבירו כיצד באמצעות שילוב של שתי פקודות אלו ניתן להדליק את הדגלים OF וCF. שימו לב במידה ואחד הדגלים כבר דלוק יש להשאירו דלוק כלומר, בסיום התהליך על שני הדגלים להיות דולקים. אין לשנות את שאר הביטים בריגסטר הדגלים. בנוסף, אין לשנות אף רגיסטר שהוא לא rflags, rip, rsp (גם לא באופן זמני). (7 נקודות)

הערה: במידה ובדקתם את עצמכם באמצעות דיבגר וראיתם שנדלק גם דגל TF זה בסדר תלמדו בהמשך מדוע הוא נדלק תוך כדי דיבוג.

**נדחוף את רגיסטר הדגלים למחסנית באמצעות הפקודה pushfq, נבצע פקודת or עליו עם הערך 0x801 כתוצאה מהפקודה, הכל ישאר כמו מקודם חוץ מביטים מס' 0 ו 11(OF ו CF) שיידלקו (במקרה והיו כבויים לפני זה)**

1. הולי התחמנית רוצה לאפשר לעצמה גישה ישירה אל התקני הקלט פלט ללא צורך בקריאות מערכת. איזה שינוי **באוגר הדגלים** יכול לעזור להולי במטרתה? (4 נקודות)

**הערה: לא צריך לציין פקודה ספציפית, רק להגיד מה צריך לעשות ברמה התיאורטית**

**היא צריכה לשנות את IOPL שנמצא בביטים 12,13 כך שיהיה גדול מן ה CPL תמיד, כלומר לשנות את ביטי ה IOPL לערכן המקסימלי - 11**

1. הולי מחליטה לנסות את התעלול מסעיף א' רק שבמקום לשנות את CF וOF היא רוצה לשנות את IOPL. להפתעתה, היא לא מצליחה לשנות את הביטים הללו. הסבירו מה ההגיון בכך שהיא לא מצליחה לשנות את IOPL? התייחסולצורך בקריאות מערכת (4 נקודת)

**אם זה היה אפשרי אז כל אחד יכל לשנות את ההרשאות שלו ללא צורך בsyscall ובכך מסכן אותנו ופותח גישה למשתמשים זדוניים הרוצים לפגוע בשאר המחשב**

**הערה**: הסעיפים הבאים קשורים לפסיקות מומלץ לענות עליהם לאחר התרגול על פסיקות.

1. וולי החבר המבולבל של הולי מתלבט כיצד ניתן לחסום פסיקות תוכנה לכן הוא שואל את הולי. אילו מבין התשובות הבאות על הולי לענות לו? יש לסמן את האפשרות הנכונה וגם לנמק בקצרה (5 נקודות)

* כיבוי דגל IF באוגר הדגלים
* הדלקת דגל IF באוגר הדגלים
* שינוי CPL ל00
* לא ניתן לחסום פסיקות תוכנה.

נימוק: **לא ניתן לחסופ פסיקות תוכנה מכיוון שהמעבד הוא זה שמביא אותן, ולא ממשיך עד שלא נתפל בהן(לעומת פסיקות חומרה שחלקן ניתן לחסום)**

1. כעת נתון שוולי הצליח להגיע למצב שבו CPL שווה ל0. וולי מעוניין לחסום פסיקות חומרה שאינן מועברות דרך כניסת NMI. כיצד הוא יכול לעשות זאת? יש לסמן את האפשרות הנכונה וגם לנמק בקצרה (5 נקודות)

* כיבוי דגל IF באוגר הדגלים
* הדלקת דגל IF באוגר הדגלים
* עליו לחבר את הפסיקות לכניסת NMI ואז לכבות את דגל IF
* לא ניתן לחסום פסיקות חומרה ולכן לא יצליח.

נימוק: **NMI – פסיקות שאינן ניתנות לחסימה, אני מעוניינים לחסום את הסוג השני (maskable interrupts) ונעשה זאת בכך שנכבה את דגל ה IF**