טכניון – מכון טכנולוגי לישראל

# ארגון ותכנות המחשב

# תרגיל 1 - חלק יבש

המתרגל האחראי על התרגיל: נדב פליקוביץ.

**שאלות על התרגיל – ב- Piazza בלבד.**

הוראות הגשה:

* ההגשה בזוגות.
* על כל יום איחור או חלק ממנו, שאינו באישור מראש, יורדו 5 נקודות.
  + ניתן לאחר ב-3 ימים לכל היותר.
* הגשות באיחור יתבצעו דרך אתר הקורס.
* לכל שאלה יש לרשום את התשובה במקום המיועד לכך.
* יש לענות **על גבי טופס התרגיל** ולהגיש אותו באתר הקורס **כקובץ PDF**.
  + ניתן להקליד את התשובות במסמך ה-WORD, או לכתוב אותן על גבי גרסת ה-PDF בעזרת הטאבלט החביב עליכן. העיקר להגיש בסופו של דבר קובץ PDF לבדיקה, בכתב ברור וקריא.

# שאלה 1 – מעקב אחר פקודות:

לפניכם קטע קוד. נתון כי **הכתובת של תחילת מקטע הנתונים היא 0xDEADBEEF**. עליכם לעקוב אחר הפקודות ולרשום תוכן של נתון מבוקש במקומות שמבקשים מכם (בערכי **הקסדצימלי**).

במידה ומתבצעת פקודה לא חוקית בשלב מסוים, יש לרשום X במקום שצריך להשלים, ולהתייחס כאילו הפקודה מעולם לא נרשמה:

.global \_start

.data

arr: .short 1, 0xEA, 0x2, 0x3, 0b1010=0XA 01 00 EA 00 02 00 03 00 0A 00

b: .long 0x19 28 37 46 46 37 28 19

c: .quad 0x0404202102052021 21 20 05 02 21 20 04 04

.bss

.lcomm jk, 8

.lcomm g\_byte, 4

.text

\_start:

xor %rcx, %rcx

movl $0x2345, %ebx

movb $0, %bl

ערך rbx: 0x2300.

xor %rax, %rax

xor %rsi, %rsi

add b, %rax, %rbx

ערך rbx: X.

lea 2(arr), %rbx

ערך rbx: X.

lea (arr), %rbx

movb 3(%rbx), %al

ערך rax: 0x00 .

mov %bh, %al

xor %al, %sil

shr $5, %rsi

movw -8(%rbx, %rsi, 2), %dx

ערך dx: 0xEA.

shl $1, %rsi

movb $0x68, g\_byte

addb (%rbx, %rsi, 2), g\_byte

ערך הבית שב- g\_byte: x.

shr $6, %rax

inc %ax

ערך rax: 0x3.

mov $jk, %rcx

lea c, %rbx

movw arr+3, %ax

ror $2, %ax

ערך rax: 0x80.

xor %ax, %ax

incb %ax

ערך rax: x.

movq (%rbx), %rbx

mov $0x40, %si

dec %rcx

movl %ebx, 2(%rcx)

התוכן שבבית בכתובת jk+3 0x05

movb $78, b

ערך הבית b (הבית שb מהווה פניה אליו): 0XBE.

movq $arr, b

ערך הבית b (הבית שb מהווה פניה אליו): 0xEF.

movswq (b), %rdx

ערך rdx: 0xFFFFFFFFFFFFBEEF.

mov $0x8529, %ax

cwd

ערך rdx: 0xFFFFFFFFFFFFFFFF.

movw $-0x33, jk

idivw jk

ערך eax:0X268 . ערך edx: 0XFFFFFFE1.

movq $0x100, (b)

imul $2, b, %rdx

ערך rax: 0x268. ערך rdx: 0x200.

xor %rax, %rax

mov $0xff, %ax

mov $4, %bl

mov $30, %rdx

imulb %bl

ערך al: 0xE0. ערך dl: 0x1E.

mov $256, %ax

mov $20, %bx

mul %bx

ערך ax: 0x1400. ערך dx: 0x0.

# שאלה 2 – תרגום מC לאסמבלי:

לפניכם קטעי קוד בשפת c עליכם לתרגם כל קטע בשפת c לאסמבלי על ידי השלמת המקומות שמסומנים בקו. במידה וכל השורה מסומנת בקו עליכם להשלים את השורה איך שאתם רוצים אך עליכם להשתמש בפוקדה אחת בלבד! נתון ש-a ו-b הוגדרו כ int.

מומלץ לעבור על "אופטימיזציה אריתמטית" מתרגול 2, ולראות דוגמאות לפני המעבר על השאלה.

**הערה 1:** בשורה הרביעית הרווח אחרי lea( אינו טעות. אין להשלים שם ערך. זהו רמז (וחלק מהסינטקס).

**הערה 2:** נזכיר כי '|' בשפת C היא הפעולה or.

על מנת למנוע בלבול מסופקת לכם **דוגמה** בשורה הראשונה:

|  |  |
| --- | --- |
| קוד בשפת c | קוד אסמבלי |
| a += b; | movl b\_\_, %eax  addl \_%eax\_,\_a\_ |
| a = a / 8; | sarl \_$3\_, a |
| a = 9\*a; | movl a, %eax  lea (%eax\_, \_%eax, 8), \_%eax\_  mov %eax, a |
| b = b\*8; | movl b, %ebx  lea ( ,%ebx,8), %ebx  mov %ebx, b |
| a = b\*2 - 11 + a; | movl a, %eax  movl b, %ebx  lea -11(%eax,%ebx,2),%eax  mov %eax, a |
| a++ | incl a |
| a = 5\*a; | imul \_\_$5\_, \_\_(a)\_, %eax  mov %eax, a |
| a = a\*a\*a\*a; | movl a, %eax  mull\_\_\_%eax\_\_\_  mull %eax\_\_  mov %eax, a |
| if (a >= 0)  b = 0;  else  b = -1; | movl a, %eax  cdq  movl %edx, b |

# שאלה 3 – לולאות ומספרים:

בשאלה זו נשתמש במספרים חסרי סימן (unsigned).

בנוסף, נניח כי הוגדר משתנה n>0 שגודלו 4 בייטים ושכל ה-General Purpose Registers מכילים 0 בתחילת התוכנית (הכוונה היא לרגיסטרים שמשתמשים בהם לחישובים ולא לריגסטרים מיוחדים כמו rip או rflags)

שירה המנהלת הבכירה בסופר כתבה קטע קוד. לפניכם הקוד ששירה כתבה:

\_start:

xor %ax, %ax

mov $1, %bx

mov (n), %cx

.L1:

mov %bx, %r9w

imul %bx, %r9w

add %r9w, %ax

inc %bx

dec %cx.

test %cx, %cx

jne .L1

END:

1. נתון שבתחילת התוכנית (בעשרוני).

מה יהיה ערך רגיסטר ax בסיום קטע התוכנית (בעת ההגעה לתווית END)? כתבו את התשובה גם בבסיס דצימלי וגם בהקסדצימלי (וכתבו את כל הבתים שלו ב-hexa)?

הערך של ax עבור n=10 יהיה 385 בבסיס דצימלי ובבסיס hexadecimal הערך של ax יהיה

0x181 .

1. הסבירו במשפט מה עושה קטע הקוד

התוכנית סוכמת את כל הריבועים מ1 עד n.

1. רמזי הסטודנט החרוץ הבחין שעבור מוחזרת תשובה לא נכונה. מה הסיבה לכך?

מהו המספר הגדול ביותר שניתן לשים ב-n בתחילת הריצה, ועדיין לקבל תשובה נכונה?

רמז: desmos

עבור n=70 נקבל מס' שהרבה יותר גדול מהכמות האפשרית הניתנת לייצוג כמס' חיובי שלם ב16 ביטים, עבור n=57 נקבל תשובה נכונה, וזהו המס' הגדול ביותר אשר עדיין מאפשר תשובה נכונה מפני שזהו המספר הגדול ביותר שסכום הריבועיים עד אליו כולל ניתן לייצוג כ-16 ביטים.

**השאלה ממשיכה בעמוד הבא**

1. רמזי, שרוצה לקבל את פרס העובד המצטיין , שינה את הקוד:

\_start:

xor %ax, %ax

mov $1, %bx

mov (n), %cx

.L1:

mov %bx, %r9w

imul %bx, %r9w

add %r9d, %eax

inc %bx

dec %cx

test %cx, %cx

jne .L1

END:

אך הקוד עדיין מחזיר תשובות שגויות עבור מספרים גדולים מהמספר שמצאתם בסעיף הקודם. מדוע?

בשורה הראשונה של התוכנית רמזי איפס רק את רגיסטר %ax ולא את רגיסטר %eax ולכן יתכן שהיה בו ערך קודם ובעת חריגת התוצאה מרגיסטר %ax לרגיסטר %eax נקבל ערך שגוי.

1. השלימו את השורות הבאות, כך שיתקבל קוד **חסר לולאות** שיחזיר את אותה תוצאה ב-ax עבור כל . ניתן להשאיר שורות ריקות.

\_start:

mov (n), %ax

mov %ax, %bx # bx = n

inc %bx # bx = n+1

imul %bx, %ax # ax = n(n+1)

addw (n), %bx # bx = 2n+1

imul %bx, %ax # ax = n(n+1)(2n+1)

xor %dx, %dx # dx = 0

mov $6, r9w # r9w = 6

idivw r9w # ax = n(n+1)(2n+1)/6, dx = 0

.

END: