ארגון ותכנות המחשב

תרגיל 1 - חלק יבש

המתרגל האחראי על התרגיל: תומר כץ.

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.

Data type

Byte (b) 8

word (s) 16

Doubleword (l) 32

Quadword (q) 64

Double Quadword 128

0x100	be
0x101	ba
0x102	fe
0x103	ca

Disford

rflag - pilcs

rip - fs ...

movl $42 -20(%rbp)

control flow

Proby %rbp

שאלה 1 – מעקב אחר פקודות:

לפניכם קטע קוד. נתון כי הכתובת של תחילת **data section** היא **xDEADBEEF0**. עליכם לעקוב אחר הפקודות ולרשום תוכן של נתון מבוקש במקומות שמבקשים מכם (בערכי הקסדצימלי .(

אם הפקודה לא חוקית בשלב מסוים, יש לרשום **X** במקום שצריך להשלים, ולהתייחס כאילו הפקודה מעולם לא נרשמה. בנוסף, נמקו מה הבעיה בפקודה.

```
.global _start

.section .data
arr: .short 6, 0xEA, 0x22, 0x4B1D, 0b1010
buff: .fill 10, 2, 0x42
id: .long 0x19283746
key: .quad 0x0406282309052021

.section .bss
.lcomm a, 8
.lcomm b, 4

.section .text
_start:
  xor %rcx, %rcx
  movl $0x5432, %ebx
  movb $4, %bl

  xor %rax, %rax
  xor %rsi, %rsi
  add b, %rax, %rbx

  lea 4(arr), %rbx

  lea (buff), %rbx
  movb 4(%rbx), %al

  movb 7(%rbx), %al

  lea (arr), %rbx
  mov %bh, %al
  xor %al, %sil
  shr $5, %rsi

  movw -4(%rbx, %rsi, 2), %dx

  shl $1, %rsi
  movb $0x68, b
  addb (%rbx, %rsi, 2), b
```

ערך rbx: 0X5404

ערך rbx: X יותר מידי פרמטרים

ערך rbx: X סינטקס לא תקין

ערך rax: 0X42

ערך rax: 0x00

ערך rsi: 0X05

ערך **dx**: 0X1D

ערך הבית b (הבית שb מהווה פניה אליו): X קורס

<u>השאלה ממשיכה בעמוד הבא</u>

```
mov $0xFFFF00, %rax
shr $8, %rax
inc %ax
```

ערך rax: 0x00

```
movw arr+3, %ax
ror $2, %ax
```

ערך rax: 0x0880

```
xor %ax, %ax
incb %ax
```

ערך rax: 0x0000

```
mov $a, %rcx
lea key, %rbx
movq (%rbx), %rbx
mov $0x40, %si
dec %rcx  0x0406282309052021
movl %ebx, 2(%rcx)
```

ערך הבית 4+a (הבית ש- 4+a מהווה פניה אליו): 0x09

```
movb $78, b
```

ערך הבית b (הבית שb מהווה פניה אליו): 0x4E

```
movq $arr, b
```

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

```
movswq (b), %rdx
```

ערך rdx: 0xFFFFFFFFFFFFBEEF

```
mov $0xAAAA, %ax
cwd
```

ערך rdx: 0xFFFFFFFFFFFFFFFF

```
movw $-0x9F, a
idivw a
```

ערך **eax**: 0x89

ערך **edx**: 0xFFC1

```
movq $0x123, (b)
imul $3, b, %rdx
```

ערך rax: 0x89

ערך rdx: 0xFFC1

```
xor %rax, %rax
mov $0xfc, %ax
mov $4, %bl
mov $015, %rdx
imulb %bl
```

ערך **al**: 0xF0

ערך **dl**: 0x0

```
leaq $0x40FE67, %rdx
```

ערך rdx: אין השמה ל מחוץ לכתובת

שאלה 2 – תרגום מC לאסמבלי:

לפניכם קטעי קוד בשפת c עליכם לתרגם כל קטע בשפת c לאסמבלי על ידי השלמת המקומות שמסומנים בקו. אם כל השורה מסומנת בקו עליכם להשלים את השורה בכל דרך שתרצו, אך עם פקודה אחת בלבד!

נתון ש-a ו-b הוגדרו כ int. מותר לכם להשתמש בכל רגיסטר עזר שתרצו.

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 not.

על מנת למנוע בלבול מסופקת לכם דוגמה בשורה הראשונה:

קוד בשפת **c**	קוד אסמבלי
`a += b;`	`movl b, %eax` `addl %eax, a`
`a = a / 16;`	`sarl $4, a`

`a = 3*a;`	`movl a, %eax`
	`lea (%eax, %eax, 2), %ebx`
	`mov %eax, a`

`b = b*8;`	`movl b, %ebx`
	`lea ( ,%ebx, $8 ), %ebx`
	`mov %ebx, b`

`if (a >= 0)`
`  b = 0;`
`else`
`  b = -1;`
	`movl a, %eax`
	`cdq`
	`movl %edx, b`

`a = b*2 - 24 + a;`	`movl a, %eax`
	`movl b, %ebx`
	`lea -24(%eax, %ebx, 2), %eax`
	`mov %eax, a`

`a--`	`decl (a)`

`a = ~(1<<16)`	`mov $0x10000, %eax`
	`not %eax`
	`mov %eax, a`

`a = a*a*a*a*a*a*a*a;`	`movl a, %eax`
	`mul %eax, %eax`
	`mul %eax, %eax`
	`mov %eax, a`

שאלה 3 – לולאות ומספרים:

בשאלה זו נשתמש במספרים חסרי סימן (unsigned).

בנוסף, נניח כי הוגדר משתנה $n > 0$ שגודלו 16 ביט ושכל ה-General Purpose Registers מכילים 0 בתחילת התוכנית (הכוונה היא לרגיסטרים שמשתמשים בהם לחישובים ולא לרגיסטרים מיוחדים כמו rip או rflags)

קורנליוס האיום כתב את קטע קוד הבא:

```
_start:
  xor %ax, %ax
  mov $1, %bx
  mov (n), %cx

.L1:
  mov %bx, %r9w
  imul %bx, %r9w
  imul %bx, %r9w
  add %r9w, %ax
  inc %bx
  dec %cx
  test %cx, %cx
  jne .L1
END:
```

1. נתון שבתחילת התוכנית $n = 10$ (בעשרוני).

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

$\sum_{i=1}^{10}(i^3) = \sum_{i=1}^{n}(i^3) = \frac{n^2(n+1)^2}{4} = 521B = 21,019$

exa

2. איזו נוסחה/ביטוי מתמטי מחשב קטע הקוד הנ"ל?

מתקבל את הסכום - $\sum_{i=1}^{n}(i^3) = \frac{n^2(n+1)^2}{4}$ התוצאה נשמרת ב- ax

3. יהודית שבאה לבקר את קורנליוס שמה לב שעבור $n = 55$ מוחזרת תשובה לא נכונה. מה הסיבה לכך? מהו המספר הגדול ביותר שניתן לשים ב-n בתחילת הריצה, ועדיין לקבל תשובה נכונה?

הסיבה לכך היא גלישה מ- ax, מספר הבתים של ax היו 2

נחשב n מקסימלי נגדיר: $\sum_{i=1}^{n}(i^3) = \frac{n^2(n+1)^2}{4} = f(n)$, 2 ביטים ← $2^{16} = 65,536$

$\frac{n^2(n+1)^2}{4} - 65\,536 = 0 \rightarrow n < 64 \rightarrow 16$

dex

השאלה ממשיכה בעמוד הבא

4. סיוון, האויבת של יהודית, רצתה להראות שהיא הכי טובה. לכן הציגה את הקוד שלה לפתרון הנוסחה:

```
_start:
  xor %rax, %rax
  mov $1, %bx
  mov (n), %cx

.L1:
  mov %bx, %r9w
  imul %bx, %r9w
   imul %bx, %r9w
   add %r9w, %eax
  inc %bx
  dec %cx
  test %cx, %cx
  jne .L1
END:
```

ענו על סעיף 3 שוב, הפעם בהתייחס לקוד של סיוון.

- עבור ערכים הגבוהים מ- 4 בתים לא יהיה ניתן לאכסן מ- CX, בנוסף eax לא מאותחל ולכן עלול להכיל ערך לבל.

$n_{max} \approx 130k$ ← $f(n) \le 2^{32}$, עבור תוצאה הגבולה מ- 4 bytes נקבל שגיאה

אם n בגודל 16 bit לכן לכל היותר n יהיה 65,536, לכן לא מכאן תבוא השגיאה אלא מ- n עצמו ולכן $n = 65536$ המקסימלי

5. השלימו את השורות הבאות, כך שיתקבל קוד חסר לולאות שיחזיר את ב-**rax** את התוצאה של הנוסחה מסעיף 2 בצורה נכונה לכל n חסר סימן בגודל 16 ביט. כמובן הניחו כי n מוגדר לכם מראש ב-section אחר ואין צורך להגדירו. ניתן להוסיף שורות, אך קוד עם יותר מ-5 פקודות יקבל ניקוד חלקי בלבד.

```
_start:
```

נשתמש בנוסחא:

$$\sum_{i=1}^{n}(i^3) = \frac{n^2(n+1)}{4}$$

```
  movl (n,1,1), %ebx
  imul n, %ebx
  imul %eax, %ebx
  imul %ebx, %ebx
  sal $2, %eax

END:
```