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.
- תיקונים בקובץ ממורקרים.

שאלה 1 – מעקב אחר פקודות:

לפניכם קטע קוד. נתון כי הכתובת של תחילת **data section** היא **0xDEADBEEF**. 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
.lcomm c, 10


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

ערך rbx: _____

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

ערך rbx: _____

```
  lea 4(arr), %rbx
```

ערך rbx: _____

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

ערך rax: _____

```
  movb 7(%rbx), %al
```

ערך rax: _____

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

ערך rsi: _____

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

ערך **dx**: _____

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

ערך הבית b (הבית שb מהווה פניה אליו): _____

השאלה ממשיכה בעמוד הבא

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

ערך rax: _____

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

ערך rax: _____

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

ערך rax: _____

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

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

```
movb $78, b
```

ערך הבית b (הבית שb מהווה פניה אליו): _____

```
movq $arr, b
```

ערך הבית b (הבית שb מהווה פניה אליו): _____

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

ערך rdx: _____

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

ערך rdx: _____

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

ערך **eax**: _____

ערך **edx**: _____

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

ערך rax: _____

ערך rdx: _____

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

ערך **al**: _____

ערך **dl**: _____

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

ערך rdx: _____

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! בתאים עם כמה שורות קוד חייבים למלא את כולן.

נתון ש-a ו-b הוגדרו כ int וכל הרגיסטרים מאותחלים ל-0. 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 _____, _____`

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

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

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

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

`a--`	`_____`

`a = ~(1<<16)`	`_____`
	`_____`
	`mov %eax, a`

`a = a*a*a*a;`	`movl a, %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)?

_____

_____

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

_____

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?

_____

_____

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 %r9d, %eax
  inc %bx
  dec %cx
  test %cx, %cx
  jne .L1
END:
```

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

_____

_____

_____

5. השלימו את השורות הבאות, כך שיתקבל קוד <u>חסר לולאות</u> 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:

  ____________________

  ____________________

  ____________________

  ____________________

  ____________________

END:
```