ש' גלדיימן 211641162

אלעד מרגוזן 206202905

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

  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
```

rcx = 0
ebx = 0x5432
bl = 4
0

rax = 0
rsi = 0

arr + 6
rsi = ..1010
b = 0x0000006B

ערך rbx: 0x5404

ערך rbx: X add לא מוגדר ב אוברנדים

ערך rbx: X שיבת מיקון לא תקינה

ערך rax: 0x42

ערך rax: 0

ערך rsi: 0x5

ערך **dx**: 0x4B1D

ערך הבית b (הבית שb מהווה פניה אליו): X לא אפשר לעשות mem mem add

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

rax= 0 ·FFFF00
rax= 0 ·FFFF

ערך rax: 0

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

ערך rax: 0x880

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

ערך rax: X אם אינו בגודל ג'מ

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

incb לא תאפס לגמרי
rcx= *a
rbx= *key
rbx=key
si=0x0040
rcx=(*a)-1

a= 0x0000090020210000

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
```

ax=0xAAAA

ערך rdx: 0xFFFFFFFFFFFFFFFF

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

ערך **eax**: 0x89

ערך **edx**: 0xFFFF FFC1

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

ערך rax: 0x89

ערך rdx: 0x369

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

rax=0
rax=fc
bl=4
rdx=0xD=13

4·13

ערך **al**: 0xF0

ערך **dl**: 0x0D

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

ערך rdx: Immediates don't have an effective adress X

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 $4, a`

`a = 3*a;`	`movl a, %eax`
	`lea (%eax, %eax 2), %eax`
	`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 $0xFFFEFFFF`
	`nop`
	`mov %eax, a`

`a = a*a*a;`	`movl a, %eax`
	`imul %eax %eax`
	`imul %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      ax=0
  mov $1, %bx       bx=1
  mov (n), %cx      cx=*n

.L1:
  mov %bx, %r9w     r9w=bx
  imul %bx, %r9w    r9w=bx^2
  imul %bx, %r9w    r9w=bx^3
  add %r9w, %ax     ax+=bx^3
  inc %bx           bx++
  dec %cx           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)?

3025

0x0BD1

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

$$\sum_{i=1}^{n} i^3$$

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?

22 הסיבה שצה לא נכון כי צה גולש

בגלל שאבר לשגולות אx% (16 בית)

הערך שלו $\sum_{i=1}^{n}$ קורס יותר מ16 ביטים ל"כם.

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      rax=0
  mov $1, %bx         bx=1
  mov (n), %cx        cx=n

.L1:
  mov %bx, %r9w       r9w=bx
  imul %bx, %r9w      r9w=bx^2
  imul %bx, %r9w      r9w=bx^3
  add %r9d, %eax      eax+=bx^3
  inc %bx
  dec %cx
  test %cx, %cx
  jne .L1
END:
```

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

כי משיר את עוקר את מה היינבאה k bx^3 אלשת מעבר

למקום k שמא, לכן מה הוא המסבר הבגדול ריוקר שיתן תשובה נכונה

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:
  mov (n), %rax
  mul %rax, %rax
  add (n), %rax
  shr %rax
  mul %rax, %rax

END:
```