הטכניון – מכון טכנולוגי לישראל

ארגון ותכנות המחשב

תרגיל 1 - חלק יבש

המתרגל האחראי על התרגיל: איתי אילת.

שאלותיכם במייל בעניינים מנהלתיים בלבד, יופנו רק אליו.

כתבו בתיבת **subject**: יבש 1 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.

מגישים: יאן טומסינסקי (207231267)
תום סמולין (313552739)

שאלה 1 (34 נק') – מעקב אחר פקודות:

לפניכם קטע קוד. נתון כי הכתובת של תחילת מקטע הנתונים היא 601038 בהקסדצימלי 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

.section .data
array: .long 0xf415, 0xf3561768, 0x8200f645
a: .short 1
b: .quad 0x670081b521c

.section .bss
.lcomm stam, 4

.section .text
_start:
xor %rcx, %rcx
xor %rax, %rax

xor %esi, %esi
lea array, %rbx
```

ערך rbx: _____0x0000000000601038____.

```
movb 3(%rbx), %al
```

ערך al: __0x00__.

```
inc %esi
mov 3(%rbx, %rsi, 8), %ecx
```

ערך ecx: _____0x1c000182_____.

```
movw (%rbx, %rsi, 2), stam
```

ערך stam (כל ה4 בתים): _____x_____.

```
xor %eax, %eax
dec %eax
```

ערך rax: _____0x00000000FFFFFFFF_____.

```
addq %rax, 2(%rbx, %rsi, 4)
```

התוכן שבכתובת 601040x0_____0x45____,

```
mov $b, %rdi
```

ערך rdi: _____0x0000000000601046____.

```
mov $2, a
```

ערך הבית a (הבית שa מהווה פניה אליו): ____x_____.

```
movq $array, a
```

ערך הבית b (הבית שb מהווה פניה אליו): ____0x60____.

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

ערך rdx: _____0x0000000000000060_____.

```
cdq
```

ערך rdx: _____0x00000000FFFFFFFF _____.

```
movl $2, a
idivl a
```

ערך eax: _____0x00000000_____. ערך edx: _____oxFFFFFFFF_____.

```
mulb %edx
```

ערך eax: _____x_____.

```
movq $1, (a)
imul $2, a, %rdx
```

ערך rdx: _____0x0000000000000002_____.

```
xor %rax, %rax
mov $0xff, %ax
mov $2, %bx
mov $1, %dx
imulb %bl
```

ערך ax: _____0xFFFE_____. ערך dx: ____0x0001_____.

שאלה 2 (32 נק') – 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 שa וb הוגדרו כ int.

מומלץ לעבור על תרגול 2 שאלה 3 ולראות דוגמאות לפני תחילת השאלה.

הערה: בשורה הרביעית הרווח אחרי lea(הוא לא טעות. אין להשלים שם ערך. זה חלק מהסינטקס.

על מנת למנוע בלבול מסופקת לכם דוגמא בשורה הראשונה:

קוד בשפת **c**	קוד אסמבלי
a += b;	movl ___b__, %eax addl ____%eax__, __a____
a = a / 4;	__sarl__, __$2__, ____a__

a = 9 * a;	movl a, %eax
	lea (__%eax_ , __%eax___, 8_), %eax
	
	mov %eax, a

a = a*4;	movl a, %eax
	
	lea (, __%eax, ___4__), %eax
	
	mov %eax, a

a = 4a + b + 9;	movl a, %eax
	
	movl b, %ebx
	
	lea 9(%ebx, %eax, 4), %eax_
	
	mov %eax, a

a++;	____incl a_____

a *= a	movl a, %eax
	
	____mull a_____
	
	mov %eax, a

a = 4a	imul _$4___, ___a____, %eax mov %eax, a

`if(a >= 0)`
`  b = 0;`
`else`
`  b = -1;`
	movl a, _%eax _____
	
	sarl $31, %eax
	
	movl ___%eax ___, b

שאלה 3(34 נק') – מעקב אחר לולאות:

בשאלה זו נשתמש במספרים חסרי סימן.

בנוסף נניח כי הוגדר משתנה res שגודלו 4 בתים ושכל הרגיסטרים מלבד ax וbx מכילים 0 בתחילת התוכנית (הכוונה היא לרגיסטרים שמשתמשים בהם לחישובים ולא לריגסטרים מיוחדים כמו rip או rflags) צ'נדלר סטודנט בקורס כתב קטע קוד. לפניכם הקוד שצ'נדלר כתב:

```
chandler_prog:
    cmp %bx, %ax
    je end
    cmp %bx, %ax
    ja a
    sub %ax, %bx
    jmp chandler_prog
a:
    sub %bx, %ax
    jmp chandler_prog
end:
    mov %ax, res
```

1. נתון שבתחילת התוכנית ax=56 וbx=42 (ערכים בעשרוני). מה יהיה ערך משתנה res (בעשרוני) בסיום קטע התוכנית (כל ארבעת הבתים) (5 נקודות)
2. הסבירו במשפט אחד מה עושה התוכנית (8 נקדות)
3. מוניקה, סטודנטית חרוצה שאוהבת סדר ודיוק במיוחד, טוענת שיש שורה מיותרת בתוכנית של צ'נדלר (זאת אומרת שאם נמחק אותה התוכנית לא תשתנה). מה השורה המיותרת? (6 נקודות)
4. מוניקה מחליטה לכתוב קוד מסודר יותר מהקוד של צ'נדלר שתפקידו לעשות אותו דבר בדיוק. לפניכם הקוד שכתבה:

```
Monica_prog:
    mov %bx, %cx
    div %bx
    mov %cx, %ax
    mov %dx, %bx
    cmp $0, %bx
    jne Monica_prog
    mov %ax, res
```

1. מצאו קלט שבו הפלט (res) של התוכנית של מוניקה זהה לפלט של התוכנית של צ'נדלר, אך מתקיים $ax \neq bx$. (2 נקודות)
2. מצאו תנאי הכרחי ומספיק, עבורו הפלט (res) של התוכנית של מוניקה זהה לפלט (res) של התוכנית של צ'נדלר. (6 נקודות)

5. הציעו תיקון או הוספה של שורה אחת בתוכנית של מוניקה על מנת שהקוד יתן את אותה תוצאה כמו הקוד של צ'נדלר. (7 נקודות).

תשובות:

1. Res = 0x0000000E
2. הקוד מבצע את אלגוריתם אוקלידס למציאת הGCD.
3. שני הcmp בקוד זהים. לאחר הcmp מבוצע עדכון של רגיסטר הדגלים. הדגלים שמעניינים אותנו ומשתנים בחישוב הcmp הם ה Arithmetic Flags (לטובת je ו ja).
ב je לא מבוצע חישוב אריתמטי (פשוט בודקים את מצבו של ZF) ולכן אף אחד מהדגלים שמעניינים אותנו לא משתנים. ולכן נוכל ישר לעבור ל ja עבור אותו החישוב הראשון. כלומר, ה cmp השני מיותר.

4. 1. ax = 28, bx = 14
2. תנאי: הערך של ax הוא כפולה של bx.
התנאי מספיק, כיוון שלפי הקוד שלעיל נקבל איטרציה אחת (השארית תהיה אפס. השארית עוברת לbx ולכן לא נקפוץ בשורה הלפני אחרונה). נשים לב שבאיטרציה זו, ax תקבל את הערך המקורי של bx וזה הערך שיכנס לres. בפרט, זהו הGCD.
התנאי הכרחי כיוון שאם ax אינה כפולה של bx, נקבל שישנה שארית שונה מ-0. השארית לפי מבנה הפקודה עוברת ל dx שאת תוכנו אנחנו מעבירים לbx. כשהשארית שונה מ-0, אנחנו חוזרים לתחילת התוכנית. נשים לב שבאיטרציה הבאה לא מבוצע איפוס של dx. ולכן למעשה החישוב div %bx שקול לחישוב div %dx, כלומר: יתבצע החישוב dx:ax / dx.
במקרה כזה, המנה ax מקיימת $ax \geq 2^{16}$. כדי לייצג ערך כזה נדרשים לפחות 17 ביטים, בעוד שב ax ישנם 16. ולכן נקבל גלישה והתוכנית תקרוס.

5. נוסיף לאחר השורה הראשונה ולפני ביצוע פקודת החילוק:
.mov $0, %dx
כדי לוודא שההרחבה של ax בחלוקה תהיה מורכבת מאפסים בלבד ובכך תמנע הגלישה שציינו לעיל.