ענו על כל השאלות, **במקום המוקצב לכך בלבד**.

חלק א' (80%)

על שאלות "אמריקאיות" (Multiple choice) סמנו **בצורה ברורה** עיגול סביב האות הנכונה, 5 נקודות לתשובה נכונה. על תשובה לא נכונה 1- נקודות. על תשובה חסרה תקבלו 0 נקודות.

שאלה 1

במחשב 80486, מספר הגישות לזכרון הנדרש לבצוע הפקודה הבאה:

```
add     eax, [X]
```

ובהנחה שה-OPODE ללא אופרנדים אורכו 2 BYTES, במקרה הגרוע ביותר (כולל FETCH) יהיה:

א) 1 ב) 4 ג) 5 ד) 8

שאלה 2

נתון קטע הקוד דלהלן (בצורת listing). מה מחזירה פונקציה Foos (ערך מוחזר ב-eax), אם ערכו של eax לפני הקריאה לפונקציה הוא 25?

```
        section .data
Foos:
        mov     ebx, [Fob]
        mov     [Foa], bx
        add     eax, eax
Foa:
        dec     eax
        call    Foos
Fob:
        ret
        mov     eax, 0x4050ea50
```

א) 0x4050ea50 ב) 10 ג) לא מחזירה כלום כי זו רקורסיה אינסופית. ד) 9

שאלה 3

נתון הקוד הבא (עם אלפבית טרנרי):

DATA WORD	CODE WORD
00	011
01	212
10	022
11	100

בעזרת קוד זה ניתן:

א) לתקן שגיאה אחת **או** לגלות שתי שגיאות.
ב) לתקן שתי שגיאות **או** לגלות שלוש שגיאות.
ג) לגלות שגיאה אחת, אבל לא מאפשר תיקון שגיאות.
ד) לשחזר עד 3 ביטים שערכם נמחק.

שאלה 4

סטודנט הציע, כדי לשפר את הקוד, להשתמש באותו קוד, אבל לרשום כל ספרה ב- 2 bits, כלומר במקום 0 נרשום 00, במקום 1 נרשום 01, ובמקום 2 נרשום 10, ונקבל מילות קוד בינאריות של 6 bits.

א) הסטודנט צדק, הקוד החדש יאפשר תקון יותר שגיאות מהקוד הקודם.
ב) הסטודנט טעה, כי זה בדיוק אותו קוד כמו קודם.
ג) בקוד החדש ניתן לתקן שתי שגיאות.
ד) בקוד החדש ניתן לתקן לכל היותר מחיקה אחת.

יתבצע בצורה נכונה (ע"י קריאה ל-Doit):

a)
```
Junk:   db   "NO", 10, 0
Doit:   mov  eax, 4       ; write
        mov  ebx, 1
        mov  ecx, Junk
        mov  edx, 4
        int  0x80
        ret
```

b)
```
Doit:   mov  eax, 4       ; write
        push dword, 0x0A4F4E
        mov  ebx, 1
        mov  ecx, esp
        mov  edx, 4
        int  0x80
        add  esp, 4
        ret
```

c)
```
Junk:   db   "NO", 10, 0
Doit:   push Junk
        call printf
        add  esp, 4
        ret
```

d)
```
Doit:   mov  eax, 4       ; write
        mov  ebx, 1
        mov  ecx, 0x0A4F4E
        mov  edx, 4
        int  0x80
        ret
```

שאלה 10

במחשב מבוסס על 86x80 ישנו UART שבו רגיסטר RECEIVE לקליטת CHARACTER מהתקשורת הטורית, ורגיסטר TRANSMIT לשליחת נתונים לתקשורת הטורית. שני הרגיסטרים נמצאים ב- MEMORY MAPPED IO בכתובת 1010H. נדרש לקרא תו המתקבל בתקשורת - ולשלוח אותו בתקשורת כפי שהתקבל. מהוא הקוד שלא יבצע זאת נכון?

1) OUT 1010H, BYTE [1010H]

2) MOV AL, [1010H]
   MOV [1010H], AL

3) OR BYTE [1010H], 0
4) AND BYTE [1010H], FFH

שאלה 11

במערכת מחשב 80486, כדי לספק הגנה, על החומרה לא לאפשר למשתמש לבצע:

א) שנוי מצב המכונה ישירות למוד SYSTEM, תוך כדי המשך ביצוע קוד משתמש.
ב) יכולת שינוי INTERRUPT VECTORS
ג) גישה לרגיסטרים SI ו-DI
ד) גישה ישירה להתקני-DMA
ה) צריך למנוע את כל האפשרויות א, ב, ד, אבל ג מותר
ו) יש למנוע ביצוע כל הסעיפים א-ד כדי להבטיח הגנה
ז) ניתן להגן בעזרת תוכנת מערכת ההפעלה בלבד, ללא כל עזרה מהחומרה.

שאלה 12

עלינו לייצר רשימה של מספרים, בה אבר אחד שערכו נתון ב-eax. (נשתמש בפונקציה malloc, שהיא פונקצית ספריה C-callable, שמקבלת מספר, ומחזירה אזור זכרון בגודל זה). ה-dword הראשון בכל רשומה הוא מצביע לרשומה הבאה, והשני מכיל את הנתון. מצביע שערכו 0 הוא null pointer. הקוד הנכון הוא:

a)
```
push eax
push 8
call malloc
add  esp, 4
pop  ebx
mov  [ebx], 0
mov  [ebx+4], eax
```

b)
```
push eax
push 8
call malloc
pop  ebx
pop  ebx
mov  [eax], 0
mov  [eax+4], ebx
```

c)
```
push 8
call malloc
add  esp, 4
mov  [eax], 0
mov  [eax+4], eax
```

d)
```
push eax
push 8
call malloc
add  esp, 4
pop  ebx
mov  [eax], eax
mov  [eax+4], eax
```

שאלה 13

נתונה הגדרת macro הבאה, וכן נתונים בזכרון כדלהלן:

```
%macro    print     3
     pushad
     mov  eax, 4      ; write
     mov  ebx, %1     ; file descriptor
     mov  ecx, %2     ; address
     mov  edx, %3     ; byte count
     int  0x80
     popad
%endmacro
section   .rodata
File:     dd   1
Frobozz:  db "Beat it", 10, 0
Frobozz_end:
```

איזה מהשימושים הבאים במקרו יגרום לפעולה לא נכונה של התוכנית:

a)
```
print     1, Frobozz, Frobozz_end-Frobozz
```

b)
```
print     [File], Frobozz, Frobozz_end-Frobozz
```

c)
```
mov  ebx, Frobozz_end
sub  ebx, Frobozz
print     1, Frobozz, ebx
```

d)
```
mov  edx, Frobozz
print     1, edx, Frobozz_end-Frobozz
```

שאלה 14

נתון הקוד הבא, עבור מעבד Motorola 68000 שעובד בשיטת Big-endian

שאלה 15

ההוראה int 0x80 גורמת לתוכנית מחשב על PENTIUM, עם LINUX:

א) גורם תמיד ל-segmentation fault.

ב) התוכנית תמיד מסיימת את הריצה עקב כך בצורה תקינה.

ג) תלוי בערך הרגיסטר eax.

ד) כל התשובות הקודמות נכונות.

ה) אף אחת מהתשובות הקודמות אינן נכונות.

שאלה 16

עלינו לממש קריאה לפונקציה בשם blah שנכתבה ב-C, עם ארגומנט אחד שערכו 0x100FA500 . סמנו את הקוד שלא יבצע זאת נכון:

a)
```
sub  esp, 2
mov  word [esp],0xA500
sub  esp, 2
mov  word [esp],0x100F
call blah
add  esp, 4
```

b)
```
push 0x0100FA500
call blah
add  esp, 4
```

c)
```
sub  esp, 4
mov  dword [esp],0x0100FA500
call blah
add  esp, 4
```

d)
```
sub  esp, 2
mov  word [esp],0x100F
sub  esp, 2
mov  word [esp],0xA500
call blah
add  esp, 4
```

חלק ב' (20%)

שאלה 1 (5%)

ב-AX נמצא הערך **0x000F**. רשום 5 פקודות שונות שכל אחת מהן יגרמו לכך שברגיסטר AL יהיה הערך 1-, ללא שמוש ב-MOV.