

ארגון המחשב ושפת סף – תרגיל 2  
סמסטר ב', תש"פ  
צבי מלמד

**מועד הגשה:** עד יום מוצ"ש 16/5/20 (grace: עד יום ראשון בבוקר)  
הנחיות הגשה:

- **קבוצות עבודה:** יש לבצע את התרגיל לבד. אין הגשה משותפת!
- **אופן הגשה:** תיבת הגשה ב MAMA.
- יש להגיש קבצי מקור בלבד. לכל שאלה בתרגיל יש לשלוח קובץ `asm`.  
בלבד של הפתרון וכן את קבצי ה-`data` שעושים להם `include` מתוך התכנית. כלומר, **אין להגיש את תוצרי הקומפילציה** (למשל קבצי `.exe`).  
במקום שמצוין יש להגיש גם את קובץ הנתונים
- שמות הקבצים צריכים להיות: `ex1_q1.asm`, `ex1_q2.asm`, `ex1_q3.asm`.
- את כל קבצי ההגשה יש לארוז בקובץ `.ZIP`.
- יש לרשום את השם ותעודת הזהות כהערה בקוד בראש כל קובץ ה-`asm`.
- בנוסף, בתחילת כל תכנית יש לכתוב לפלט שורה שכוללת את שמכם, תעודת הזהות, ומספר השאלה – לדוגמה:  

```
==>ex1_q1.exe
Tzvi Melamed id: 012345678 Ex1-Q1
```
- יש לוודא שהתוכנית שאתם שולחים עוברת קומפילציה ומצליחה לרוץ – **שאלה שלא עוברת קומפילציה, תקבל ציון 0.**
- **קבצי `data`:**
  - במקרה שנדרש להגיש בנפרד קובץ של מקטע הנתונים, שמו של הקובץ יהיה במתכונת `ex1_q1_data.inc` (למשל, עבור תרגיל #1 שאלה #1).
  - יש לצרף קובץ זה לקובץ ה-`ZIP` שאתם מגישים.
  - קובץ זה יכול הגדרות נתונים בהתאם למוגדר בשאלה. **התכנית צריכה לעבוד נכון עם ערכים שונים של הנתונים.** כלומר, **בודק התרגיל, עשוי להשתמש בקובץ דומה, אבל עם ערכים שונים**, והתכנית (כמובן) צריכה לעבוד נכון.
  - במקרה שדרושים לכם נתונים נוספים, **שאינם מוגדרים בשאלה**, למשל, משתנים לתוצאות ביניים, **עליכם להגדיר קטע `data` נוסף, בתוך קובץ ה-`asm` שלכם.** כמו כן, מחרוזת ההדפסה שכוללת את שמכם ותעודת הזהות, צריכה להיות בתוך ה-`asm`. ולא בתוך קובץ ה-`data` שעושים לו `include`.

- קווים מנחים, לתשומת לבכם, המשפיעים על הניקוד:
1. יש לשים לב להוראות בנוגע להגדרת משתנים, העברת פרמטרים, התנהגות מבוקשת של התוכניות והפונקציות ומבנה הפלט שאמור להתקבל.
  2. למען קריאות התרגיל יש לתעד בתחילת התרגיל מה התרגיל עושה. בתוכנית עצמה יש להקפיד על כתיבת הערות באנגלית (במקומות שמתבקש. הפעילו שיקול דעת).
  3. כל הגשה באיחור מצריכה את אישור המרצה, ומראש.
  4. איחור בהגשה ללא אישור: איחור בהגשה עד שבוע מפחית מהציון הסופי של התרגיל 10%. איחור מעבר לשבוע, מפחית 25%.
  5. התוכנית צריכה לעבור קומפילציה ולהיות נכונה מבחינת syntax ולעבוד נכון!.
  6. על התוכניות להישלח כ - Source Code משמעות הדבר הינה, שיש לשלוח את קבצי ה **ASM בלבד** ולא את שאר התוצרים. כפי שצוין לעיל.
  7. בתרגילים שבהם מוגדרים קטעי נתונים – אין להסתפק בבדיקת נכונות עם הנתונים שמופיעים (מוגדרים) בקטע הנתונים הנתון. **עליכם לבדוק את נכונות הפתרון שלכם עם נתונים נוספים, בהתאם לתנאי השאלה.**

#### פונקציות הספרייה:

בכל התכניות עליכם להשתמש בספריית הפונקציה של Irvine Kip. תוכלו למצוא אותה בסוף ספר הלימוד, או שתגלו משהו כמו Irvine kip library function.

## שאלה #1 - כתיבת משולש לטרמינל

עליכם לכתוב תכנית שמדפיסה משולש בגודל מסוים שמורכב מתו כלשהו (ראו למטה צילום מסך של דוגמת הרצה).

תחילה התכנית מבקשת מהמשתמש להקליד תו כלשהו והיא קוראת אותו.

לאחר מכן, התכנית מבקשת מהמשתמש להכניס מספר (גודל המשולש).

לאחר מכן התכנית מדפיסה את המשולש ומסתיימת.

להלן צילום המסך של ההרצה:

```
==>ex2_q1
Tzvi Melamed id: 012345678 Ex2-Q1
Please Enter a character: &
Please Enter the triangle size: 6
    &
   &&
  &&&&
 &&&&&&
&&&&&&&&
&&&&&&&&&&
==>
```

## הנחיות והקלות:

- את התו עליכם לקלוט ע"י קריאה לפונקציה `readChar`. הפונקציה הזאת קולטת תו מבלי לבצע `echo` שלו למסך. ולכן, מיד לאחר קליטת התו, בצעו את הקריאה לפונקציה `writeChar`. כמו כן, מיד לאחר מכן, עליכם לקרוא לפונקציה `CRLF`.
- את המספר N עבור גודל המשולש עליכם לקלוט ע"י קריאה לפונקציה `readDec`.
- הקלה: מותר להדפיס את המשולש כפי שמצויר, כלומר גם השורה התחתונה מתחילה בתו אחד רווח, ובמקרה כזה, השורה הראשונה תחיל N רווחים. (מדוע זה קצת יותר קל? כי אתם לא צריכים לטפל בלולאה שמתבצעת אפס פעמים).
- הודעות ההדפסה – מופיעות להלן. בצעו העתק-הדבק לקטע קוד הזה:

```
header    BYTE "Your Name id: 012345678 Ex2-Q1",10,13,0
rqst_char BYTE "Please Enter a character: ",0;
rqst_size BYTE "Please Enter the triangle size: ",0;
```

## שאלה #2 סכום וקטורים (מערכים)

נתונים ארבעה וקטורים (מערכים), כולם בעלי אותו אורך (כלומר אותו מספר איברים).  
 המערך Bvec הוא מערך של מספרים **לא-מסומנים** בגודל BYTE.  
 המערך Wvec הוא מערך של מספרים **מסומנים** בגודל WORD.  
 המערך Dvec הוא מערך של מספרים **מסומנים** בגודל DWORD.  
 באזור הנתונים עליכם להכריז על מערך SumVec של DWORD שאיננו מאותחל, באורך זהה, והוא נועד להכיל את הסכום של שלושת הווקטורים, **כמספרים מסומנים**.

עליכם לכתוב תכנית שסוכמת את האיברים של שלושת הווקטורים לתוך sumVec.

כלומר:

```
SumVec[j] = Bvec[j] + Wvec[j] + Dvec[j];
```

שימו לב שבתהליך הזה, יש להמיר את הערכים בווקטור Bvec שאיננו מסומן, לערכים מסומנים. כלומר, אם קיים שם איבר בגודל בית, שערכו הוא 0FFh, אזי הוא יומר לערך +255.

לאחר שהתכנית סוכמת את כל הערכים לתוך הווקטור SumVec, היא מדפיסה הודעה מתאימה (ראו בדוגמא) ולאחר מכן, הדפסת המערך על ידי קריאה לפונקציה Print\_Dword\_Arr שמופיעה בהמשך.

התכנית שלכם תבצע include לקובץ בשם: ex2\_q2\_data.inc. קובץ זה יכיל את מקטע הנתונים הבא (וכן את הקוד של הפונקציה שמוזכרת בהמשך). הקובץ נמצא במאמא.

```
.data
Bvec    BYTE    0FFh, 0FEh, 10, 3 ; 255, 254, 10, 3
Wvec    SWORD   -5, -54, 100, 3000
Dvec    SDWORD  -500, 200, 300, 3000000
SumVec  SDWORD  lengthOf Bvec DUP (?)
msg1    BYTE    "Sum Vector is:", 10, 13, 0
```

עבור מקטע זה, הפלט של התכנית יהיה:

```
==>ex2_q2
Tzvi Melamed id: 012345678 Ex2-Q2
-250 +400 +410 +3003003
==>
```

## הנחיות:

- התכנית צריכה לעבוד גם עם נתונים אחרים, למשל, מערכים בעלי שם זהה, אותו טיפוס byte, word or dword, אך בגודל שונה (כמות שונה של איברים). **הבהרה:** הסיבה לכך היא שבדוק התרגילים יכול להחליף את המערכים בקובץ שנתון לכם, בקובץ שמכיל מערכים אחרים, והתכנית צריכה לעבוד נכון. בכל אופן, בכדי שהתכנית תעבוד, המערכים שבהם הוא ישתמש יהיו בעלי אותו שם ואותו סוג של משתנה).
- במידה ואתם צריכים הגדרות נוספות או מקטע נתונים נוסף (לדוגמא מחרוזת ההדפסה כל הכותרת כולל השם שלכם) עליכם להגדיר אותם בתוך קובץ התכנית שלכם (ולא בקובץ ה include).

- השתמשו בפונקציה writeInt להדפסת המספרים.
- השתמשו ב"מעט אוגרים".
- השתדלו לכתוב קוד קצר .
- בחרו שיטת (שיטות) מיעון אופטימלית לבעיה הנתונה.
- השתמשו בפונקציה הבאה (שנמצאת בתוך הקובץ הנתון ex2\_q2\_data.inc) בכדי להדפיס את המערך. עליכם להציב את הערכים המתאימים לתוך האוגרים, ואז לכתוב קריאה לפונקציה ע"י הפקודה: `call Print_Dword_Arr`.

```
; Print_Dword_Arr: prints signed array of dwords
; Getting arguments in registers:
;   EDI - holds the address of the array (first element)
;   ECX - holds the number of elements
Print_Dword_Arr PROC
    push eax
    push ecx
    push edi
L1:
    mov eax, [edi]
    call WriteInt
    mov al, ' '
    call writeChar
    add edi, 4
    loop L1

    call CRLF
    pop edi
    pop ecx
    pop eax
    ret
Print_Dword_Arr ENDP
```

### שאלה #3 סינון תווים לא-אלפאביתיים ממחרוזת והמרה ל- Upper-Case

עליכם לכתוב תכנית שמבצעת את ההתנהגות להלן.

תחילה התכנית קולטת מחרוזת מהקלט. ניתן להניח שאורכה קטן מ-80 (זה מתבצע באמצעות הקריאה ל `readString`). התכנית מסננת מהמחרוזת תווים שאינם אלפאביתיים. את התווים שהם אותיות קטנות של האלפאבית, היא ממירה לאותיות גדולות.

לאחר מכן, התכנית מדפיסה את המחרוזת.

להלן דוגמת הרצה של התכנית:

```
==>ex2_q3
Tzvi Melamed id: 012345678 Ex2-Q3
Enter a string:
hello !#Good@%&-Nigh-1t!!
After processing the string is:
HELLOGOODNIGHT
==>
```

בהצלחה!!