

ארגון המחשב ושפת סף – תרגיל 2

סמסטר ב', 2016

צבי מלמד

הנחיות הגשה:

- מועד הגשה: **עד ליום שישי 13/5/16** בחצות (חצות שבין יום שישי לשבת).
- **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. כלומר, אין להגיש את תוצרי הקומפילציה (למשל קבצי **.exe**). במקום שמצוין יש להגיש גם את קובץ הנתונים שעושים לו **include** מתוך התכנית.
 - שמות הקבצים צריכים להיות: **ex2_q1.asm, ex2_q2.asm**.
 - את כל קבצי ההגשה יש לארוז בקובץ **.ZIP**.
 - יש לרשום את השם ותעודת הזהות כהערה בקוד בראש כל קובץ ה- **asm**.
 - **יש לוודא שהתוכנית שאתם שולחים עוברת קומפילציה ומצליחה לרוץ – שאלה שלא עוברת קומפילציה, תקבל ציון 0.**
- **צילומי מסך:**
 - לכל שאלה עליכם לצרף צילום מסך של החלון שבו קימפלתם והרצתם את התרגיל. לשם כך השתמשו ב **ALT+Print-screen** או בכלי **snipping tool** של **WINDOWS**.
 - אין לשלוח צילום של כל המסך!
 - שמות הקבצים של צילומי המסך צריכים להיות כמו קבצי ה **ASM** רק עם סיומת **png**, למשל: **ex2_q1.png, ex2_q2.png**.
- **קבצי data:**
 - במקרה שנדרש להגיש בנפרד קובץ של מקטע הנתונים, שמו של הקובץ יהיה במתכונת **ex2_q1_data.inc** (למשל, עבור תרגיל #2 שאלה #1). יש לצרף קובץ זה לקובץ ה **ZIP** שאתם מגישים.
 - קובץ זה יכיל 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. בעיקר בתחילת כל פונקציה ובמקום שמתבקש (כלומר לא ברור) – יש להבהיר את הכוונה. כל הגשה באיחור מצריכה את אישור המרצה, ומראש.
3. איחור בהגשה ללא אישור: איחור בהגשה עד שבוע מפחית מהציון הסופי של התרגיל 10%. איחור מעבר לשבוע, מפחית 25%.
4. התוכנית צריכה לעבור קומפילציה ולהיות נכונה מבחינת syntax ולעבוד נכון!.
5. על התוכניות להישלח כ - Source Code משמעות הדבר הינה, שיש לשלוח את קבצי ה **ASM בלבד** ולא את שאר התוצרים. כפי שצוין לעיל, יש לצרף גם צילומי מסך של ההרצה.
6. התוכניות צריכות להיות קצרות ולעניין – Keep it simple.
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.**

שאלה #1 סכום ווקטורים (מערכים)

בשאלה זאת כל המספרים הם מסומנים.

נתונים שני ווקטורים (מערכים) אחד בגודל WORD והשני בגודל DWORD. המערכים באורך זהה. באזור הנתונים, עליכם להגדיר מערך שלישי בשם SumVec מטיפוס DWORD באורך זהה לשניים האלו.

עליכם לכתוב תכנית שמבצעת את שתי המשימות הבאות:

א. מסכמת את האיבר במקום j בשני המערכים, וכותבת את הסכום לאיבר j במערך SumVec. שימו לב שתוך כדי כך התכנית ממירה את נתוני המערך בגודל WORD לערכים בגודל DWORD.

כלומר, אם נסמן את הווקטורים בשמות: SumVec, Vec1, Vec2, אזי:

$$\text{SumVec}[j] = \text{Vec1}[j] + \text{Vec2}[j]$$

ב. התכנית מחשבת את **סכום כל האיברים** בשני המערכים, וכותבת את התוצאה למשתנה Sum. כלומר יוצרת העתק של המערך הזה.

בסיום ההרצה של התכנית היא מדפיסה את ווקטור הסכום ואת סכום הווקטורים, כפי שמודגם להלן:

התכנית שלכם תבצע include לקובץ בשם: **ex2_q1_data.inc**. קובץ זה יכיל את מקטע הנתונים הבא:

```
.data
Vec1  SWORD  10, -50, 100, 200
Vec2  SDWORD -500, 200, 300, 7
SumVec  SDWORD lengthOf Vec1 DUP (?)
Sum  SDWORD ?
msg1  BYTE  "Sum = ",0
msg2  BYTE  "Sum Vector is:", 10, 13, 0
```

עבור מקטע זה, הפלט של התכנית יהיה:

```
$==> ex2_q1.exe
Tzvi Melamed id: 012345678 Ex2-Q1
Sum = +267
Sum Vector is:
-490 +150 +400 +207
```

הנחיות:

- במידה ואתם צריכים הגדרות נוספות או מקטע נתונים נוסף (לדוגמא מחרוזת ההדפסה כל הכותרת כולל השם שלכם) עליכם להגדיר אותם בתוך קובץ התכנית שלכם.
- השתמשו בפונקציה `writeln` להדפסת המספרים.
- השתמשו ב"מעט אוגרים".

- השתדלו לכתוב קוד קצר .
- בחרו שיטת (שיטות) מיעון אופטימלית לבעיה הנתונה.
- השתמשו בפונקציה הבאה (צרפו אותה לקוד) בכדי להדפיס את המערך:

```
; 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 edx
L1:
    mov eax, [EDX]
    call WriteInt
    mov al, ' '
    call writeChar
    add edx, 4
    loop L1

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

הרחבה לתרגיל (רשות, לא להגשה)

הרחיבו את התכנית שלכם, כך שתתמוך במערכים באורך שונה. כלומר, Vec1, Vec2, עשויים להכיל מספר שונה של איברים. במקרה כזה, ההקצאה של Vec3 צריכה להיות באורך המקסימלי מבין שניהם. הקטע הבא מבצע את ההקצאה הנכונה של Vec3: (אגב... כל הקטע הזה כתוב ב data.. הביטויים השונים כולל IF וכו' הם רק בזמן אסמבלי).

```
Vec1  SWORD    10, -50
Vec2  SDWORD   -500, 200, 300, 7

IF ((LengthOf Vec1) GT (LengthOf Vec2))
MAX_LEN = LengthOf Vec1
ELSE
MAX_LEN = LengthOf Vec2
ENDIF

SumVec      SDWORD MAX_LEN DUP (?)
```

שאלה #2: סינון ומיון

עליכם לכתוב תכנית שמבצעת סינון מחרוזות + מיון בועות (על המחרוזות המסוננות). התכנית משתמשת בפונקציות. העברת הארגומנטים בכל הפונקציות מתבצעת באמצעות אוגרים.

התרגיל כתוב בשלבים – מומלץ מאוד לעקוב אחר השלבים. להריץ ולדאבג כל שלב, ורק לאחר מכן לעבור לשלב הבא. בוודאות, שזה יחסוך לכם זמן בסופו של דבר. אגב, כך מפתחים תכנה, לא רק באסמבלר, אלא בכל שפה, ולא רק בלימודים, אלא גם בתעשייה.

שלב א'

כתבו פונקציה בשם `ls_alpha`. הפונקציה מקבלת באוגר `AL` ערך כלשהו, ומחזירה באוגר `EAX` את הערך 1 אם הערך ב- `AL` מייצג אות (גדולה או קטנה) ואחרת מוחזר הערך 0.

שלב ב'

כתבו פונקציה בשם `filter_string`. הפונקציה מקבלת באוגרים את הארגומנטים הבאים:

- (1) מצביע למחרוזת המקור (כלומר, כתובת/offset ההתחלה של המערך)
- (2) מצביע למחרוזת המטרה (לאחר הסינון)

הפונקציה סורקת את מחרוזת המקור, ומעתיקה למחרוזת המטרה רק את התווים שהם אותיות (גדולות או קטנות).

שלב ג'

עליכם לכתוב פונקציה בשם `sort_string` שממיינת את המחרוזות (זאת שיצרנו בשלב ב') מהקטן אל הגדול. הפונקציה הזאת מקבלת באוגרים (א) את כתובת המחרוזת שאותה היא צריכה למיין (ב) את אורך המחרוזת.

ראשית מומלץ שתעשו מיון שהוא case-sensitive. לאחר מכן תשפרו את המיון כך שיהיה case-insensitive – זאת היכולת שנדרשת בסופו של דבר (את זה עליכם להגיש).

אתם רשאים לכתוב פונקציית עזר שמשווה בין שני תווים ולהיעזר בה.

מחרוזת המקור מוגדרת בקובץ `ex2_q2_data.inc`. כלומר, התכנית שלכם תבצע `include` לקובץ בשם: `ex2_q2_data.inc`. קובץ זה יכיל את מקטע הנתונים שמתואר למטה:

במידה ואתם צריכים הגדרות נוספות או מקטע נתונים נוסף (לדוגמא מחרוזות להדפסה) עליכם להגדיר אותם **בתוך קובץ התכנית שלכם**.

הנחיות:

- בפונקציה המיון `sort_string` **אין להשתמש בפקודה Loop**. כלומר.. מן הסתם, כידוע, דרושה לולאה, ואפילו מקוננת בכדי לממש מיון בועות. אבל בתרגיל הזה, עליכם לממש את הלולאה על ידי קפיצות מותנות וניהול עצמאי של המונים.
- מומלץ לממש פונקציה `STRLEN` שמקבלת מצביע למחרוזת שמסתיימת ב `NULL` ומחזירה באוגר `EAX` את אורך המחרוזת. (התו `NULL` איננו נספר). הערך שיתקבל כאן יעבור כארגומנט האורך לפונקציה `Sort_String`

מקטע הנתונים בקובץ **ex2_q2_data.inc** יכיל את הטקסט הבא:

```
.data
SrcStr  BYTE  "HELLO !Good-Bye?? @$%ZYX", 0
DestStr BYTE  Lengthof SrcStr DUP(?)

msg1    BYTE  "The String Before:",10,13,0
msg2    BYTE  "The String After Filtering:",10,13,0
msg3    BYTE  "The String After Sorting:",10,13,0
```

עבור הנתונים האלו, הפלט של התכנית צריך להיות כמו כאן:

```
$==> ex2_q2.exe
Tzvi Melamed id: 012345678 Ex2-Q2
The String Before:
HELLO !Good-Bye?? @$%ZYX
The String After Filtering:
HELLOGoodByeZYX
The String After Sorting:
BdEeGHLL0ooXyYZ
```

בהצלחה!!