

ארגון המחשב ושפת סף – תרגיל 1

סמסטר א', תשע"ט
צבי מלמד

מועד הגשה: עד יום חמישי 6/12/18 **בחצות** (בחצות שבין חמישי לשישי) [ההגשה נדחתה בכמה ימים בגלל השיעור החסר / שביתת הסטודנטים]

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**.
 - **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: ex1_q1.png, ex2_q2.png, ex1_q3.png
- **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 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.
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, יש לצרף גם צילומי מסך של ההרצה.
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: כפל ללא שימוש בפקודה MUL

כתבו תוכנית המחשבת מכפלה של שני מספרים. בשאלה זאת אין להשתמש בפקודה MUL או IMUL אלא בלולאה שמבצעת סכום מספרים (ADD).

- הנכפל – הינו מספר מסומן בגודל של מילה (WORD).
- הכופל – הינו מספר לא מסומן בגודל של בית (BYTE).
- המכפלה – תהיה מסומן במספר בגודל של מילה כפולה (DWORD).
- יש להגדיר משתנים בזיכרון עבור הכופל, הנכפל והתוצאה.
- עבור הנכפל והכופל – יש להגדיר ערכים.
- התוצאה תישמר בזיכרון במקום של המכפלה (במשתנה).

עליכם לבצע include לקובץ נתונים ששמו ex1_q1_data.inc
קובץ זה יכיל את ההגדרות הבאות:

```
.data
MSG BYTE "The program performed this calculation:",10,13,0
M WORD -2500
N BYTE 11
RESULT DWORD ?
```

התוכנית תדפיס את הפלט הבא:

```
==>ex1_q1.exe
Tzvi Melamed id: 012345678 Ex1-Q1
The program performed this calculation:
-2500 * +11 = -27500
==>
```

הערות

- השתמשו בפונקציה writeInt בכדי להדפיס מספרים מסומנים (עם הסימן + או - לפני המספר).
- כזכור, הפלט אמור להיות דומה (אם לא זהה פרט לשם התלמיד) לפלט הנ"ל.
- לגבי השימוש בפונקציה writeInt או פונקציה "אחות" writeDec – ראו את ההגדרות בספר ו/או את ההערה שמופיעה בשאלה הבאה. (השמת ערך לתוך EAX).

שאלה 2: מספרי טריבונצ'י

"נמצא" סדרת מספרים שנקרא לה Tri-Bonachi – טריבונצ'י שמוגדרת באופן הבא:

$$t(n) = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ t(n-1) * 2 + t(n-2) & \text{all other cases} \end{cases}$$

כלומר, האיברים ברשימה יהיו 1, 1, 3, 7, 17, 41, 99, ...

הקבוע (קבוע אסמבלי) N=10 מגדיר כמה מספרים עליכם לחשב.

הנחיות:

- השתמשו בטיפוס DWORD עבור ערכי טריבונצ'י.
- התכנית מדפיסה את המספר הראשון ואת המספר השני לפני ביצוע הלולאה.
- התכנית נכנסת ללולאה... מבצעת אותה ובכל איטרציה מדפיסה מספר טריבונצ'י הנוכחי.
- בכדי להדפיס מספר לא-מסומן עליכם להשתמש בפונקציה WriteDec. הפונקציה הזאת מדפיסה את המספר שנמצא באוגר EAX. לפיכך, לפני שקוראים לה, צריך להעביר את הערך הדרוש ל EAX ורק אז לקרוא לה.
- עליכם להדפיס מפריד בין המספרים – כלומר פסיק ואחריו רווח. (ראו את קובץ הנתונים להלן).
- בתחילה התכנית, עליכם להדפיס שורה שמכילה את שמכם ואת מספר הסטודנט שלכם. וכמו כן כותרת Ex1-Q2
- לאחר אחר הדפסת הסדרה יש לרדת שורה ולהדפיס את סכום איברי הסדרה עד כה עם כותרת מתאימה (ראו דוגמת פלט)
- בסוף התכנית עליכם להדפיס New-Line (אתם יכולים להשתמש בקריאה לפונקציה CRLF).

עליכם לבצע include לקובץ נתונים ששמו: ex1_q2_data.inc.
קובץ זה יכיל את ההגדרות הבאות:

```
N = 10;
.data
separator BYTE ", ", 0
msg BYTE "The sum of the Tri-Bonachi is: ", 0
```

להלן דוגמא של המסך שמתקבל (כאשר N=10):

```
==>ex1_q2.exe
Tzvi Melamed id: 012345678 Ex1-Q2
1, 1, 3, 7, 17, 41, 99, 239, 577, 1393
The sum of the Tri-Bonachi is: 2378
==>
```

הערות:

- ניתן להניח $N \geq 3$
- ניתן להניח שאברי הסדרה או סכום הסדרה, לא חורגים מהמספר הלא מסומן המקסימלי שניתן לייצג ב-32 ביט.

שאלה 3: לוח הכפל

עליכם לכתוב תכנית שקולטת מספר שלם N מהקלט (ע"י קריאה לפונקציה readInt). לאחר מכן, התכנית מדפיסה את לוח הכפל בגודל NxN.

הנחיות:

- אין להשתמש בפקודה mul או imul שאותן לא למדנו.
- אפשר להניח שהמספר שנקלט הוא "קטן וסביר" - למשל מספר קטן מ-20.
- מומלץ להשתמש במשתנים בעלי שמות משמעותיים (אפשר לכתוב את התכנית תוך שימוש באוגרים בלבד, אבל יהיה לכם יותר קל אם תשתמשו במשתנים).

להלן דוגמא להרצת התכנית:

```
==>ex1_q3.exe
Tzvi Melamed id: 012345678 Ex1-Q3
Enter a number: 7
Multiplication Table is:
1 2 3 4 5 6 7
2 4 6 8 10 12 14
3 6 9 12 15 18 21
4 8 12 16 20 24 28
5 10 15 20 25 30 35
6 12 18 24 30 36 42
7 14 21 28 35 42 49

==>
```

להלן תוכן הקובץ ex1_q3_data.inc :

```
.data
MSG1 BYTE "Enter a number: ", 0
MSG2 BYTE "Multiplication Table is: ", 10, 13, 0
```

בהצלחה!!