

ארגון המחשב ושפת סף – תרגיל 3 סמסטר ב', תש"פ צבי מלמד

מועד הגשה: עד יום ראשון בחצות 14/6/20 (grace: עד יום שני)

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 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 (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 חישוב סכום מערכים והשוואה

נתונים שני מערכים של **טיפוס WORD**. הם יכולים להיות בגודל (מספר איברים) כלשהו. המערכים נקראים בשמות המפתיעים: Arr_1, Arr_2. הפונקציה הראשית, מחשבת את סכום האיברים בכל אחד מהמערכים, ומשווה ביניהם. בהתאם היא כותבת הודעה, איזה מהמערכים הוא בעל סכום האיברים הגדול יותר. המערכים כמו גם המחרוזות להדפסה ניתנים בקובץ ex3_q1_data.inc. את חישוב סכום המערכים מבצעים באמצעות פונקציה, שיודעת לחשב את סכום האיברים במערך. עליכם להחליט על חתימת הפונקציה (כלומר, איזה ארגומנטים היא מקבלת). החזרת הסכום שהיא חישבה, מתבצעת כמקובל, באמצעות האוגר EAX. **הפונקציה מקבלת את הארגומנטים במחסנית.**

להלן תכן הקובץ ex3_q1_data.inc ולאחריו, דוגמת הרצה:

```
.data
Arr_1 WORD 100, 200, 300, 1000, 2000, -3000
Arr_2 WORD 10000, -2000, 20000, 10000, 30000, 1, 2, 3
msg1 BYTE "Sum of Arr_1 is: ", 0
msg2 BYTE "Sum of Arr_2 is: ", 0
arr1_bigger BYTE "Sum of Arr_1 is Bigger than sum of Arr_2", 10, 13, 0
arr2_bigger BYTE "Sum of Arr_2 is Bigger than sum of Arr_1", 10, 13, 0
```

עבור הנתונים במקטע הזה, הפלט של התכנית יהיה:

```
==>ex3_q1
Tzvi Melamed id: 012345678 Ex3-Q1
Sum of Arr_1 is: +600
Sum of Arr_2 is: +68006
Sum of Arr_2 is Bigger than sum of Arr_1
==>
```

הנחיות

- א. ניתן להניח שהמערכים לא ריקים (לפחות איבר אחד בכל מערך).
- ב. פתחו (כלומר, כיתבו ובידקו) את התכנית בשלבים.
 - I. בשלב ראשון, בדקו שהפונקציה שמחשבת את סכום המערך עובדת היטב. לשם כך מספיק לקרוא לה עבור אחד המערכים בלבד. **זאת פונקציה מאוד פשוטה – תרגיל כניסה מעולה לעולם הפונקציות.** שימו לב שהפונקציה הזאת מחשבת סכום האיברים, שכ"א מהם הוא WORD אבל הסכום צריך להיות DWORD (וכך גם הערך המוחזר).
 - II. לאחר מכן, וודאו ששתי הקריאות לפונקציה עובדות היטב.
 - III. בשלב האחרון, הוסיפו את ההשוואות וההדפסות הנחוצות.

שאלה #2 - כתיבת משולש לטרמינל

כפי שתראו מיד – הלוגיקה בשאלה הזאת, דומה, או אפילו זהה, למה שהיה בתרגיל הקודם. אתם רשאים כמובן להשתמש בקוד של התרגיל הקודם. התוספות העיקריות בתרגיל הזה, קשורות לעבודה של פונקציות – כלומר, לחומר שלמדנו בשלושת השיעורים האחרונים.

א. עליכם לכתוב פונקציה בשם `print_triangle_1` שמקבלת **שני ארגומנטים בגודל בית באוגרים**: (1) תו ASCII (2) מספר כלשהו (לא מסומן). הפונקציה מציירת משולש בגודל המספר (ארגומנט #2) של התו (ארגומנט #1).

ב. עליכם לכתוב פונקציה בשם `print_triangle_2` שמקבלת **שני ארגומנטים בגודל WORD במחסנית**: (1) תו ASCII (2) מספר כלשהו (לא מסומן). אפשר להניח שהוא קטן מ 255. לגבי כ"א מהארגומנטים האלו, הפונקציה תתעלם מהבית ה MSB, ותשתמש רק בבית ה LSB. הפונקציה תדפיס משולש שמורכב מהתו המדובר, ע"כ שתקרא לפונקציה מסעיף א.

ג. הפונקציה הראשית מתנהגת כמו שאלה #1 בתרגיל הקודם. כלומר:

תחילה התכנית מבקשת מהמשתמש להקליד תו כלשהו והיא קוראת אותו.

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

הפונקציה הזאת תכתוב את המשולש למסך, ע"י קריאה לפונקציה מסעיף ב', כלומר קריאה ל- `print_triangle_2`.

הנחיות

ההנחיות וגם הפלט הם כמו בתרגיל הקודם.

שאלה #3- כתיבת משולש הפוך ע"י פונקציה רקורסיבית

בתרגיל זה עליכם לכתוב קוד באסמבלר, שדומה לתכנית הבאה בשפת C:

```
#include <stdlib.h>
#include <stdio.h>

#define SIZE 8
#define CHAR '$'

void write_triangle_rec(int size, char c, int org_size)
{
    if(size == 0)
        return;
    for (int j = 0; j < org_size - size; j++)
        printf(" ");
    for (int j = 0; j < size + size; j++)
        printf("%c", c);
    printf("\n");
    write_triangle_rec(--size, c, org_size);
}

int main()
{
    write_triangle_rec(SIZE, CHAR, SIZE);
}
```

להלן צילום המסך של ההרצה:

```
=>ex3_q3
Tzvi Melamed id: 012345678 Ex3-Q3
$$$$$$$$$$$$
$$$$$$$$$$$$
$$$$$$$$$$$$
$$$$$$$$$$$$
$$$$$$$$
$$$$$$$
$$$$$
$$$
$$
```

הנחיות:

- בתכנית זאת, עליכם להדפיס את המשולש **החל מהמקום השמאלי ביותר בחלון**.
- הקובץ ex3_q3_data.inc יכיל את שתי ההגדרות הבאות:


```
.data
SSIZE = 8
CHAR = '$'
```
- שימו לב שהמילה SIZE היא מילה שמורה באסמבלר, ולכן בקובץ ההגדרות מופיעה המילה SSIZE.
- העברת הארגומנטים היא במחסנית! כל הארגומנטים (גם התו) נדחפים כ DWORD.**
 - סדר העברת הארגומנטים, כפי שמקובל (מה שלמדנו בשיעורים - מימין לשמאל).
 - אל תשתמשו במשתנה לוקלי עבור ז' (המונה בלולאות). התשתמשו ב ECX במקום.
 - השתמשו בשמות סימבוליים לשמות הארגומנטים.
 - תוך כדי תהליך הפיתוח, אני ממליץ לכם, במקום תו רווח בתחילת השורה, להדפיס תו אחר, למשל קו-תחתון. זה עשוי לעזור לכם בדיבוג!

בהצלחה!!