

תרגיל בית 3 – הגשה ב-09/03/2024

הערות להגשה:

- את העבודה יש להגיש בזוגות בלבד.
- יש להגיש את התרגיל אך ורק מהמודל של **מאחד מבני הזוג** (אל תגישו שניכם בנפרד!).
- יש להגיש:
 - קובץ hw3.asm, ושם יהיה **כל** הקוד שלכם. כולל הסברוטינות שבתרגילים קודמים מימשתם בקבצים נפרדים. בתרגיל זה, **כל** הקוד יהיה בקובץ אחד. זה יהיה קובץ ארוך!
 - קובץ submitters.txt עם הת"זים ואחוז ההשתתפות בהתאם לפירוט בהמשך.
- עליכם לציין בקובצי ה-asm וה-txt את תעודות הזהות ואת אחוז העבודה של כל שותף, באופן הבא בלבד:
 - ; 123456, 30%
 - ; 456789, 70%
- שימו לב לרווחים, פסיקים ונקודה פסיק שבתבנית לעיל. אל תחרגו ממנה!
- יתרה מזאת, בקובץ ה-asm, אין לבצע מרווח של שורה או יותר בין שורות הת"ז לבין שורת ה-orig.
- כפי שתראו בדוגמה המצורפת בעמוד האחרון של התרגיל.
- **שימו לב, שותף אשר לא יהיה פעיל בתרגיל הבית עלול להיקנס בנקודות.**
- **תהיה השגחה על כך ואף תיתכן בדיקה מדגמית של הבקאות שלכם בפתרון.**
- **ניתן להגיש באיחור של עד ארבעה ימים ללא עדכון צוות הקורס. כל יום איחור יגרור 5 נקודות הורדה בציון.**
- **לא תתקבלנה הגשות לאחר ארבעה ימים ממועד ההגשה.**
- אנא הקפידו על פתרון נקי, אלגנטי ומתועד לכל אורכו, אשר יסביר את הלוגיקה שלכם עם שמות לייבלים משמעותיים. שימו לב – זהו קוד אסמבלי, לכן תיעוד טוב אינו רשות, אלא **חובה!**
- על שמות הלייבלים של הסברוטינות/פונקציות שלכם להיות **בדיוק**, אחד לאחד, כפי שנכתב בתרגיל זה, יש לשים לב ל-case sensitivity של הלייבלים (באופן כללי).
- שימו לב, הטסטים נבדקים אחד אחרי השני ללא איפוס הסימולטור. **זה אומר תוספת עבודה בתרגיל זה!**
- **בדיקת תרגילי הבית הינה חצי אוטומטית. אי עמידה בדרישות הנ"ל עשויה לגרור הורדה בציון!**
 - אי עמידה בדרישות הפורמט אשר כולל:
 - שורות תעודת הזהות
 - שמות הקבצים
- **תגרור הורדה של נקודות מסך הציון של תרגיל הבית.**
- הדפסות של התרגיל צריכות להיות בהתאם לתבנית המוצגת בתרגיל.
- **כל שורת פלט אשר אינה זהה לאחד לפלט המצופה תגרור הורדה של נקודה מהטסט.**
- שאלות בנוגע לתרגיל הבית יש להפנות לפורום.
- בעניינים אישיים ניתן לפנות למתרגל במייל:

AlonPapiniCSTA@gmail.com

שימו לב – בתרגיל זה לא תקבלו קובץ קוד התחלתי, ולמעשה תצטרכו לכתוב ידנית גם את מבני הנתונים (רשימות מקושרות, מערכים) שאיתם תעבדו!

אנא קראו היטב את ההוראות!

בתרגיל זה נתרגל עבודה עם רשימות מקושרות ב-LC3, כאשר כל הקוד, המערכים והרשימות בקובץ אחד.

באוניברסיטה קיים מרצה חרוץ אשר מלמד 3 קורסים שונים.

ברצונו לבדוק מי הם 6 הסטודנטים המצטיינים מכל הסטודנטים שלו, בהתאם לממוצע הציונים של הסטודנטים. לשם כך המרצה רוצה לחשב את הממוצע על סמך 3 תרגילי הבית האחרונים ובוחן אמצע (לכולם יש משקל שווה). עליכם לתחזק 3 רשימות מקושרות דו-כיוונית של שלושת הקורסים השונים, כאשר בכל קורס קיימים לכל היותר 10 סטודנטים. כל תא ברשימה מייצג סטודנט בודד ע"י 5 פרמטרים: ציוני 3 תרגילי הבית, הבוחן וממוצע. להלן:

```
struct Student{
    Int hw1;
    Int hw2;
    Int hw3;
    Int middle_Test;
    Int average;
    Student *previous;
    Student *next;
}
```

(1) ממשו סברוטינה בשם **GetStudentGrades** – המקבלת ברגיסטר R1 כתובת לרשימה מקושרת של הקורס הרלוונטי וב-R2 את מספר הסטודנטים בקורס, עבורם יש לקלוט את 4 הציונים, ערכים בין 0 ל-100. הציונים מופרדים על ידי תו רווח אחד או יותר, כל שורה חדשה מייצגת סטודנט חדש.

(2) ממשו סברוטינה בשם **AverageCalculator** – המקבלת ברגיסטר R1 כתובת לרשימה המקושרת של הקורס הרלוונטי וב-R2 את מספר הסטודנטים בקורס, על הסברוטינה לחשב עבור כל סטודנט את הממוצע של ציוני הסטודנט ולעדכן את השדה המתאים לכל סטודנט ברשימה. הממוצע הינו מספר שלם (לא צריך שארית).

(3) ממשו סברוטינה בשם **BubbleSort** – המקבלת ברגיסטר R1 כתובת לרשימה מקושרת וב-R2 את אורך הרשימה, הסברוטינה ממיינת את הרשימה מהגדול לקטן באמצעות האלגוריתם מיון בועות **על פי הציון הממוצע של כל סטודנט**. מצורף פסאודו קוד עבור האלגוריתם בסוף הקובץ (אבל תזכרו שאתם תממשו מיון בועות על רשימות מקושרות, לא על מערך!).

(4) ממשו סברוטינה בשם **BestStudent** – המקבלת ברגיסטר R0 כתובת של מערך באורך 6, ברגיסטר R1 כתובת לרשימה מקושרת וב-R2 את אורך הרשימה, הסברוטינה מוצאת את 6 הממוצעים הטובים ביותר ברשימת הסטודנטים שב-R1 ושומרת אותם במערך השמור ב-R0 בסדר גדול לקטן (בשביל זה המיון מקודם). במידה שיש חזרות על ציון ממוצע יש להכניסו אך ורק פעם אחת למערך.

(5) כתבו קוד מעטפת המקבל מהמשתמש את מספר הסטודנטים בכל קורס, מספר הסטודנטים נע בין 1 ל-10. על הערכים להיות מופרדים בתו רווח אחד או יותר ולהסתיים בשורה חדשה. לאחר מכן, יתבקש המשתמש להכניס את ציוני הסטודנטים בכל קורס בהתאם להודעה שתודפס. כל שורה מייצגת סטודנט, ערכי הציונים מופרדים בתו רווח אחד או יותר. לסיום, יש להדפיס את 6 הממוצעים הטובים ביותר **בכל הקורסים יחדיו**. היעזרו בכל הסברוטינות שלעיל.

ניתן להניח:

- תקינות הקלט. אנחנו לא נתעסק פה בקלט לא חוקי. תניחו שאם יש קלט – הוא לפי התבנית המוגדרת.
- למרצה יש לפחות 6 סטודנטים בכל הקורסים יחדיו (אבל זה לא אומר שלא יהיה מקרה של שני ציונים ממוצעים זהים! כלומר, ייתכן שלא יהיו 6 ציונים הכי גבוהים, אז בהדפסת שאר המקומות שבמערך – 0).
- מותר להביא את הסברוטינות שמימשותם בתרגילים קודמים לקובץ הקוד של תרגיל זה, וזה ידרוש התאמה.
- תיתקלו בבעיות של מרחק PCOffset בגלל שהכל בקובץ אחד ענק. כדאי להתכונן מראש לתרחישים כאלו, ולצורך שבפקודות כמו LDI/STI, או בלהזיז סברוטינות/מערכים/רשימות כדי שלא יהיו תקלות מרחק.

להלן המחרוזות שצריכות להתלוות להדפסות של התרגיל ההדפסות (אפשר גם למצוא אותן בקבצי ה-Input/output לדוגמה שתקבלו במודל):

Enter the number of students in each course:
Enter the student grades in course 1:
Enter the student grades in course 2:
Enter the student grades in course 3:
The six highest scores are:

Enter the number of students in each course: 10 3 5

Enter the student grades in course 1:

90 75 100 86

89 56 49 60

100 100 100 100

100 90 100 98

56 75 89 100

64 75 100 86

89 56 49 60

100 100 100 100

70 90 100 98

62 75 89 100

Enter the student grades in course 2:

87 90 100 98

56 46 89 100

90 73 100 86

Enter the student grades in course 3:

100 90 100 98

90 75 100 86

89 56 49 60

64 70 80 100

56 75 89 100

The six highest scores are: 100 97 93 89 87 81

מבנה הקוד המוגש יראה כך:

; ID of student 1, __%

; ID of student 2, __%

.ORIG x3000

Write here your 'main' code!

HALT

Write here your constant/labels/arrays/lists/subroutines!

.END

פסאודו קוד – bubble sort:

נלקח מ-GeeksforGeeks

```
void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

// A function to implement bubble sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
    {
        // Last i elements are already in place
        for (j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
    }
}
```

שימו לב כי אלגוריתם זה מתבסס על מימוש על מערכים, אבל אתם עוסקים ברשימות מקושרות, אז צריך לשנות בהתאם.