

# מטלת מנחה (ממ"ן) 22

הקורס: 20465 - מעבדה בתכנות מערכות

חומר הלימוד למטלה: פרקים 4,5,6

מספר השאלות: 1 משקל המטלה: 8 נקודות (רשות)

סמסטר: 2018א' מועד אחרון להגשה: 24.12.2017

## קיימות שתי חלופות להגשת מטלות:

- שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
  - שליחת מטלות באמצעות דואר אלקטרוני, באישור המנחה בלבד
- הסבר מפורט ב"נוהל הגשת מטלות מנחה"

יש לקמפל עם דגלים מקסימליים, לקבלת כל האזהרות: **-Wall -ansi -pedantic** נדרש להגיש את קבצי המקור (.c, .h), קבצי ההרצה (את קבצי .o אין צורך לצרף), קבצי הסביבה המתאימים (כולל קבצי MAKEFILE), קבצי קלט וקבצי פלט (לפי הנחייה במפגש / באתר). נדרש ששם התיקיה ושם הקובץ לריצה יהיו כשם קובץ התוכנית הראשית, ללא הסיומת .c. יש להגיש תכניות מלאות (בין השאר מכילות main), ניתנות להידור/הרצה/בדיקת הנדרש, ללא תוספות קוד. את המטלה יש להגיש בקובץ zip. לאחר ההגשה יש להוריד את המטלה משרת האו"פ למחשב האישי, ולבדוק שהקבצים אכן הוגשו באופן תקין.

## שאלה 1 (בקבצים mat.c, mat.h, ותכנית ראשית בקובץ mymat.c)

עליכם לכתוב "מחשב כיס" אינטראקטיבי לביצוע פעולות חשבוניות על מטריצות.

תזכורת:

הפעולות החשבוניות הבסיסיות על מטריצות הן:

חיבור מטריצות:  
דוגמה:

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{array} + \begin{array}{cccc} 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 0 \end{array} = \begin{array}{cccc} 3 & 4 & 5 & 4 \\ 3 & 4 & 5 & 4 \\ 3 & 4 & 5 & 4 \\ 3 & 4 & 5 & 4 \end{array}$$

חיסור מטריצות:  
דוגמה:

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array} - \begin{array}{cccc} 0.5 & 0 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 2 & 3 \end{array} = \begin{array}{cccc} 0.5 & 1 & 0 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & -1 \end{array}$$

כפל מטריצות :

דוגמה :

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 5 & 3 & 3 & 4 \\ 10 & 6 & 6 & 8 \\ 5 & 3 & 3 & 4 \\ 10 & 6 & 6 & 8 \end{pmatrix}$$

שחלוף (transposition) של מטריצה :

דוגמה :

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ -1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & -1 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 2 & 1 & 2 \end{pmatrix}$$

כפל מטריצה בסקלר :

דוגמה :

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 2 & 3 \end{pmatrix} * 0.5 = \begin{pmatrix} 0.5 & 0 & 0.5 & 0 \\ 1 & 1 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 1 & 1.5 \end{pmatrix}$$

עליכם לכתוב תכנית הקוראת פקודות מהקלט הסטנדרטי, מפענחת ומבצעת אותן. הפקודות עוסקות בפעולות על מטריצות (על פי התזכורת למעלה).

עליכם להגדיר, תוך השימוש ב- `typedef`, את הטיפוס `mat` אשר מחזיק מטריצה בגודל 4 על 4. איברי המטריצה הם מספרים ממשיים. על מבנה הנתונים שבחרתם להיות יעיל מבחינת כמות זיכרון הנדרשת, ויעיל מבחינת הגישה אליו.

בנוסף עליכם להגדיר בתכנית הראשית שישה משתנים מטיפוס `mat`, בשמות הבאים :

`MAT_A, MAT_B, MAT_C, MAT_D, MAT_E, MAT_F`

בפקודות שלהלן, כל פרמטר שהוא שם של מטריצה יילקח מתוך ששת המשתנים האלה. בתחילת התכנית, כל המטריצות מאותחלות לאפסים

להלן הפקודות המוגדרות כקלט לתכנית :

## 1. רשימת ערכים ממשיים מופרדים בפסיקים, שם-מטריצה `read_mat`

הפקודה מציבה את הערכים שברשימה לתוך המטריצה ששמה ניתן בפקודה, לפי סדר השורות. אם ברשימה יש פחות מ-16 ערכים, האיברים שלא נתקבל עבורם ערך יכילו אפסים. אם יש יותר מ-16 ערכים, התכנית תתעלם מהערכים העודפים. חובה שיהיה ברשימה לפחות ערך אחד.

הערכים בקלט הם מספרים ממשיים בפורמט עשרוני.

לדוגמה, הפקודה : `read_mat MAT_A,5,6.25,-7,1,20`. תציב בתא (0,0) במטריצה `MAT_A` את הערך 5, בתא (0,1) את הערך 6.25, בתא (0,2) את הערך -7. בתא (0,3) את הערך 1, ובתא (1,0) את הערך 20. ביתר תאי מטריצה `MAT_A` יוצב הערך 0.

## 2. שם מטריצה `print_mat`

הפקודה מדפיסה את המטריצה ששמה ניתן בפקודה, בתצוגה דו-מימדית נאה .

יש להקפיד בהדפסה על עימוד נאה ומיושר של אברי המריצה. זכרו שמדובר במספרים ממשיים. מותר להסתפק בהדפסה עם דיוק של 2 ספרות מימין לנקודה, וכן ברוחב שדה

של 7 תווים לכל המספר (כולל נקודה-עשרונית וסימן מינוס). במידה והחלק השלם של מספר דורש רוחב שדה גדול יותר, מותרת סטיה מהעימוד המיושר. מומלץ להשתמש ביכולות של הפונקציה printf לשלוט במבנה השדה המודפס.

לדוגמה: הפקודה `print_mat MAT_A` (לאחר ביצוע הפקודה `read_mat` מהסעיף הקודם) תבצע הדפסה בסגנון הבא (או דומה לו):

5	6.25	-7	1
20	0	0	0
0	0	0	0
0	0	0	0

3. **שם-מטריצה-ג', שם-מטריצה-ב', שם-מטריצה-א' `add_mat`**

הפקודה מחברת את מטריצה א' ומטריצה ב' ומאכסנת את התוצאה במטריצה ג'.

4. **שם-מטריצה-ג', שם-מטריצה-ב', שם-מטריצה-א' `sub_mat`**

הפקודה מחסרת את מטריצה ב' ממטריצה א' ומאכסנת את התוצאה במטריצה ג'.

5. **שם-מטריצה-ג', שם-מטריצה-ב', שם-מטריצה-א' `mul_mat`**

הפקודה מכפילה את מטריצה א' במטריצה ב' ומאכסנת את התוצאה במטריצה ג'.

6. **שם-מטריצה ב', ערך-ממשי, שם-מטריצה-א' `mul_scalar`**

הפקודה מכפילה את מטריצה א' בערך ממשי (הפרמטר השני) ומאכסנת את התוצאה במטריצה ב'.

7. **שם מטריצה ב', שם-מטריצה-א' `trans_mat`**

הפקודה מבצעת היפוך (transpose) של מטריצה א', ומאכסנת את התוצאה במטריצה ב'.

8. **`stop`**

הפקודה גורמת לסיום התכנית.

**לתשומת לב:** אותו שם מטריצה יכול לשמש ביותר מפרמטר אחד באותה הפקודה. מימוש הפעולות החשבוניות על מטריצות צריך להתחשב בכך (לא לדרוס נתונים תוך כדי החישוב). לדוגמה, הפקודות שלהלן תקינות ומוגדרות היטב:

```
mul_mat MAT_A, MAT_B, MAT_A
trans_mat MAT_C, MAT_C
```

#### אופן פעולת התכנית:

יש לממש ממשק משתמש ידידותי, כך שהמשתמש יוכל להבין בכל שלב של התכנית מה עליו לעשות. בפרט, על התכנית להודיע באמצעות סימן (prompt) בכל פעם שהיא מוכנה לקלוט את הפקודה הבאה. התכנית תמשיך לקלוט ולבצע פקודה אחרי פקודה, עד שתקבל הפקודה `stop`.

**כל שורת קלט תכיל פקודה אחת מלאה, כולל כל הפרמטרים.**

מותרות גם שורות ריקות (המכילות רק תווים לבנים).

בכל פקודה, שם הפקודה מופרד מהפרמטר הראשון באמצעות רווח או טאב, ובין כל שני אופרנדים סמוכים יש פסיק.

לתשומת לב: בשורת הקלט יכולים להיות רווחים וטאבים באופן חופשי ובכמות בלתי מוגבלת, מלבד באמצע מספרים, או באמצע שמות של פקודות או מטריצות. רווחים וטאבים יכולים להיות גם בתחילת השורה ובסופה. על התכנית להתעלם מכל הרווחים והטאבים העודפים.

התכנית אינה מניחה שהקלט תקין. על התכנית לנתח כל פקודה ולוודא שאין בה שגיאות (ראו דוגמאות בהמשך). במידה ונתגלתה שגיאה, התכנית תדפיס הודעת שגיאה פרטנית, ותעבור לפקודה הבאה, בלי לבצע את הפקודה השגויה. אין לעצור את התכנית עם גילוי השגיאה הראשונה. אין צורך לדווח על יותר משגיאה אחת בכל שורת קלט.

יש לטפל גם במצב של EOF (גמר הקלט). עצירת התכנית שלא באמצעות פקודת stop אינה נחשבת תקינה (גם לא כאשר הקלט מגיע מקובץ), ויש להדפיס הודעת שגיאה על כך ואז לעצור.

להלן דוגמאות של קלט שגוי:  
שימו לב: ייתכנו סוגים נוספים של שגיאות בקלט.  
עליכם לחשוב על כל מגוון השגיאות האפשריות, ולטפל בכולן.

לפקודה:

```
read_mat MAT_G, 3.2, 8
```

יש להגיב בהודעה כגון:

```
Undefined matrix name
```

לפקודה:

```
do_it MAT_A, MAT_B, MAT_C
```

יש להגיב בהודעה כגון:

```
Unknown command name
```

לפקודה:

```
read_mat MAT_A, abc, 567
```

יש להגיב בהודעה כגון:

```
Parameter not a real number
```

לפקודה:

```
mul_mat MAT_B, MAT_C
```

יש להגיב בהודעה כגון:

```
Missing parameter
```

לפקודה:

```
trans_mat MAT_B, MAT_C, MAT_D
```

יש להגיב בהודעה כגון:

```
Extraneous text at end of command
```

לפקודה:

```
print_mat, MAT_A
```

יש להגיב בהודעה כגון:

```
Invalid comma after command name
```

לפקודה:

```
trans_mat MAT_A MAT_B
```

יש להגיב בהודעה כגון:

```
Missing comma
```

לפקודה:

```
add_mat MAT_A, , MAT_B, MAT_C
```

יש להגיב בהודעה כגון:

## Multiple consecutive commas

לפקודה :

mul\_scalar MAT\_A, 4.25, MAT\_A,

יש להגיב בהודעה כגון :

Extraneous text at end of command

להלן דוגמה של סדרת פקודות שכולן תקינות :

```
read_mat MAT_A, 1, 2, 3, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 6, 6, 6, 6, 6
read_mat MAT_B, 1, 2.3456, -7.89
read_mat MAT_C, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
print_mat MAT_A
print_mat MAT_B
print_mat MAT_C
add_mat MAT_A, MAT_B, MAT_D
print_mat MAT_D
sub_mat MAT_B, MAT_A, MAT_E
print_mat MAT_E
mul_mat MAT_B, MAT_C, MAT_F
print_mat MAT_F
mul_scalar MAT_A, 12.5, MAT_A
print_mat MAT_A
trans_mat MAT_C, MAT_C
print_mat MAT_C
stop
```

דרישות נוספות :

יש לפצל את התכנית למספר קבצים : mat.c, mymat.c, ו-mat.h.  
בקובץ mat.c יש לרכז את הפונקציות החישוביות על מטריצות. בקובץ mymat.c תהיה הפונקציה main, וכן כל פעילויות האינטראקציה עם המשתמש, ניתוח הפקודות, והדפסת התוצאות. ההצהרות (אב טיפוס) של הפונקציות מהקובץ mat.c מיוצאות באמצעות הקובץ mat.h, לשימוש הקובץ mymat.c.

באפשרותכם לבנות קבצי מקור נוספים (למשל : קובץ המכיל פונקציות עזר לניתוח הקלט, וכד').

**הקלט לתכנית הוא מ-stdin, ויכול להגיע מהמקלדת או מקובץ (באמצעות redirection בעת הפעלת התכנית).**

לפני הניתוח של כל שורת קלט, על התכנית להדפיס את השורה לפלט בדיוק כפי שנקראה. זאת כדי שניתן יהיה לראות בפלט את הפקודות המקוריות, גם כאשר הקלט מגיע מקובץ.

**חובה לצרף להגשה הרצת בדיקה**, המשתמשת בכל סוגי הפקודות, וכן מדגימה את הטיפול במגוון השגיאות השונות. מומלץ להכניס פקודת הדפסה של מטריצת התוצאה אחרי כל פקודה, כדי להראות שהתוצאה אכן נכונה (ראה לעיל הדוגמה של סדרת פקודות תקינות).  
**יש להגיש קובץ קלט + תדפיס מסך (או קובץ פלט) של הרצת הבדיקה.**

להזכירכם : לא תנתן דחייה בהגשת הממ"ן, פרט למקרים מיוחדים כגון מילואים או מחלה, במקרים אלו יש לקבל אישור הגשה מצוות הקורס.

**בהצלחה !**

