

מטלת מנחה (ממ"ן) 22

הקורס: 20465 - מעבדה בתכנות מערכות

חומר הלימוד למטלה: פרקים 4,5,6

משקל המטלה: 8 נקודות (רשות)

מספר השאלות: 1

מועד אחרון להגשה: 25.05.2025

סמסטר: 2025ב'

אופן הגשת המטלות:

שליחת מטלות באמצעות מערכת המטלות המקוונת באתר הבית של הקורס
הסבר מפורט ב"נוהל הגשת מטלות מנחה"

יש לקמפל עם דגלים מקסימליים, לקבלת כל האזהרות: -Wall -ansi -pedantic. יש להגיש את קבצי המקור c. h. אם קיימים, קבצי ההרצה (את קבצי o. אין צורך לצרף), קבצי הסביבה המתאימים (כולל קבצי makefile), וכן קבצי קלט ותדפיסי מסך או קבצי פלט (לפי ההנחיות במטלה/במפגש/באתר).

קבצי התוכנית יהיו בתיקה. נדרש ששם התיקה ושם הקובץ לריצה יהיו כשם הקובץ המכיל את הפונקציה main, ללא הסיומת c.

יש להגיש תכנית מלאה (בין השאר מכילה main), הניתנת להידור והרצה, ומאפשרת בדיקה של כל התוצאות המגוונות של הריצה ללא צורך בשינויים כלשהם בקוד המקור של התוכנית. את המטלה יש להגיש בקובץ zip.

חשוב מאוד:

- אופן הגשת המטלה והקבצים הנדרשים להגשה מופיעים כאן וכן בעמודים ה-ז בסעיף תיאור המטלות. במקרה של הנחייה שונה בפורום, יש לוודא את הנושא עם מנחה הקבוצה.
- לאחר ההגשה, יש להוריד את המטלה משרת האו"פ למחשב האישי, ולבדוק שהקבצים אכן הועלו למערכת באופן תקין.

שאלה 1 (בקבצים עיקריים mymat.c, mymat.h, ותכנית ראשית בקובץ mainmat.c)

עליכם לכתוב תכנית שפועלת כ"מחשב כיס" אינטראקטיבי לבצוע פעולות חשבוניות על מטריצות.

תזכורת:

להלן כמה פעולות חשבוניות בסיסיות על מטריצות.

חיבור מטריצות.
דוגמה :

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

חיסור מטריצות.
דוגמה :

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

כפל מטריצות.
דוגמה :

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

כפל מטריצה בסקלר.
דוגמה :

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

שחלוף (transposition) של מטריצה.
דוגמה :

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

משימות התכנית :

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

עליכם להגדיר, תוך שימוש ב- typedef, את הטיפוס mat אשר מחזיק מטריצה בגודל 4 על 4. איברי המטריצה הם מספרים ממשיים. מבנה הנתונים שהגדרתם צריך להיות חסכוני מבחינת ניצול הזיכרון, ויעיל מבחינת הגישה אליו.

בנוסף, עליכם להגדיר בתכנית הראשית (בפונקציה main) שישה משתנים מטיפוס mat, בשמות : MAT_A, MAT_B, MAT_C, MAT_D, MAT_E, MAT_F

בתחילת ריצת התכנית, יש לאתחל את כל המטריצות עם אפסים בכל האיברים.

כעת, עליכם לבצע פעולות חשבוניות על מטריצות. כל פעולה תופעל באמצעות פקודה שמועברת מהמשתמש בקלט לתכנית. בפקודות שמפורטות להלן, כל אופרנד שהוא שם של מטריצה יהיה אחד מששת המשתנים שהוגדרו לעיל. כיוון הקריאה של נוסח הפקודה הוא משמאל לימין.

מבנה הפקודות המשמשות כקלט לתכנית :

1. הצבת ערכים במטריצה :

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 בבסיס עשרוני.

לדוגמה, הפקודה: read_mat MAT_A, 5, 6.253, -7, -200.5, 23
תציב בתא [0,0] במטריצה MAT_A את הערך 5, בתא [0,1] את הערך 6.253, בתא [0,2] את הערך -7. בתא [0,3] את הערך -200.5, ובתא [1,0] את הערך 23. ביתר תאי המטריצה MAT_A יוצב הערך 0.

2. הדפסת מטריצה :

שם מטריצה print_mat

הפקודה מדפיסה את תוכן המטריצה ששמה ניתן בפקודה, בתצוגה דו-מימדית נאה. הערכים יודפסו בבסיס עשרוני.

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 לפי הצורך). במידה והחלק השלם של מספר דורש רוחב שדה גדול יותר, מותרת סטיה מהעימוד המיושר בשורה זו. מומלץ להשתמש ביכולות של הפונקציה printf לשלוט בפורמט של השדה המודפס.

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

5.00	6.25	-7.00	-200.50
23.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00

3. חיבור מטריצות :

שם-מטריצה-ג', שם-מטריצה-ב', שם-מטריצה-א' add_mat

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

4. חיסור מטריצות :

שם-מטריצה-ג', שם-מטריצה-ב', שם-מטריצה-א' sub_mat

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

5. כפל ממטריצות :

שם-מטריצה-ג', שם-מטריצה-ב', שם-מטריצה-א' mul_mat

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

6. כפל מטריצה בסקלר :

שם-מטריצה ב', ערך-ממשי, שם-מטריצה-א' mul_scalar

הפקודה מכפילה את מטריצה א' בערך ממשי (הפרמטר השני) ומאכסנת את התוצאה במטריצה ב'. הערך הממשי נתון בבסיס עשרוני.

7. שחלוף מטריצה :

שם מטריצה ב', שם-מטריצה-א' trans_mat

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

8. **סיום התכנית :**

stop

הפקודה גורמת לסיום התכנית.

לתשומת לב: אותו שם מטריצה יכול לשמש ביותר מארגומנט 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

המבנה התחבירי של הקלט :

- כל פקודה תופיע בשלמותה בשורת קלט יחידה, כולל כל הארגומנטים. מותרות גם שורות ריקות (שורות המכילות רק תווים לבנים).
- שם הפקודה מופרד מהארגומנט הראשון באמצעות רווחים ו/או טאבים (אחד או יותר).
- בין כל שני ארגומנטים יש פסיק אחד. לפני ואחרי הפסיק יכולים להיות רווחים ו/או טאבים בכמות בלתי מוגבלת. אסור שיהיה פסיק אחרי הפרמטר האחרון.
- יכולים להיות רווחים ו/או טאבים בכמות בלתי מוגבלת בתחילת השורה לפני שם הפקודה, וגם בסוף השורה (אחרי הארגומנט האחרון).
- אסור שיהיו תווים מיותרים (תווי זבל) בסוף השורה (למעט תווים לבנים).
- שמות הפקודות יופיעו באותיות קטנות בלבד, ושמות המשתנים באותיות גדולות בלבד.

אופן פעולת התכנית :

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

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). סיום התכנית שלא באמצעות פקודת stop מפורשת בקלט אינה נחשבת תקינה (גם לא כאשר הקלט מגיע מקובץ באמצעות redirection), ויש להדפיס על כך הודעת שגיאה ורק אז לעצור. לתשומת לב: השורה האחרונה בקובץ קלט אינה חייבת להסתיים בתו "שורה חדשה".

להלן דוגמאות של קלט שגוי :

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

1. לפקודה:
`read_mat MAT_G, 3.2, 8`
Undefined matrix name
יש להגיב בהודעה כגון :
2. לפקודה:
`read_mat mat_a, 3.2, -5.3`
Undefined matrix name
יש להגיב בהודעה כגון :
3. לפקודה:
`do_it MAT_A, MAT_B, MAT_C`
Undefined command name
יש להגיב בהודעה כגון :
4. לפקודה:
`Add_Mat MAT_A, MAT_B, MAT_C`
Undefined command name
יש להגיב בהודעה כגון :
5. לפקודה:
`read_mat MAT_A, abc, 567`
Argument is not a real number
יש להגיב בהודעה כגון :
6. לפקודה:
`read_mat MAT_A, 3, -4.2, 6,`
Extraneous text after end of command
יש להגיב בהודעה כגון :
7. לפקודה:
`read_mat MAT_A`
Missing argument
יש להגיב בהודעה כגון :
8. לפקודה:
`mul_mat MAT_B, MAT_C`
Missing argument
יש להגיב בהודעה כגון :
9. לפקודה:
`trans_mat MAT_B, MAT_C, MAT_D`
Extraneous text after end of command
יש להגיב בהודעה כגון :
10. לפקודה:
`print_mat, MAT_A`
Illegal comma
יש להגיב בהודעה כגון :
11. לפקודה:
`trans_mat MAT_A MAT_B`
Missing comma
יש להגיב בהודעה כגון :
12. לפקודה:
`sub_mat MAT_A, , MAT_B, MAT_C`
Multiple consecutive commas
יש להגיב בהודעה כגון :
13. לפקודה:
`mul_scalar MAT_A, MAT_B, MAT_C`

יש להגיב בהודעה כגון :

Argument is not a scalar

14. לפקודה :

stop now

יש להגיב בהודעה כגון :

Extraneous text after end of command

להלן דוגמה של סדרת פקודות שכולן תקינות :

הערה : סדרה כגון זו יכולה לשמש כקלט בהרצת בדיקה ללא טיפול בשגיאות בקלט.

```
print_mat MAT_A
print_mat MAT_B
print_mat MAT_C
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
read_mat MAT_B, 0
print_mat MAT_B
mul_mat MAT_A, MAT_A, MAT_A
print_mat MAT_A
stop
```

ארגון קוד התכנית :

יש לחלק את התכנית למספר קבצי מקור : mymat.c, mymat.h, mainmat.c

- בקובץ mymat.c יש לרכז את הפונקציות החישוביות על מטריות. לא יבוצע כל קלט/פלט בקובץ זה.
- בקובץ mainmat.c תהיה הפונקציה main, וכן כל פעילויות האינטראקציה עם המשתמש, וניתוח הפקודות (לרבות הודעות שגיאה). כמו כן, בפונקציה main יוגדרו ששת המשתנים מטיפוס mat.
- בקובץ mymat.h תהיה הגדרת טיפוס הנתונים mat, וכן ההצהרות (אב-טיפוס) של הפונקציות הממומשות בקובץ mymat.c. יש לכלול (#include) את הקובץ mymat.h בקבצי המקור האחרים.
- ניתן 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). לנוחיותכם, הכינו מספר קבצי קלט והשתמשו בהם שוב ושוב כדיבוג התכנית. בכל קובץ קלט תהיה סדרה של פקודות מגוונות על מטריות.

על התכנית להדפיס הודעת בקשה ידידותית לקלט עבור כל שורת קלט (כל פקודה). כמו כן, יש להדפיס באופן יזום מתוך התכנית את השורה כפי שנקלטה, וזאת לפני הניתוח של הפקודה.

באופן זה, שורת הקלט תוצג גם כאשר הקלט מגיע מקובץ (כידוע, נתונים הנקראים מקובץ אינם מוצגים במסך בזמן הקלט).

חובה לצרף להגשה הרצות דוגמה (אחת או יותר), המדגימות את השימוש בכל סוגי הפעולות ובכל ששת המטריצות, וכן את הטיפול בכל מגוון השגיאות בקלט.
רמז: מומלץ להכניס פקודת הדפסה של מטריצת התוצאה אחרי כל פעולה, כדי להראות שהתוצאה אכן נכונה (ראו לעיל הדוגמה של סדרת פקודות תקינות).
יש להגיש תדפיסי מסך (או קבצי פלט) של כל ההרצות. יש להגיש גם קבצי קלט.

לתשומת לבכם: לא תינתן דחיה בהגשת הממ"ן, פרט למקרים חריגים כגון אשפוז. במקרים אלו יש לבקש ולקבל אישור מראש ממנחה הקבוצה.

בהצלחה!