

# 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: 2025 א'

מועד אחרון להגשה: 12.01.2025

## 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 תוכנית 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 complex.h, complex.c, mycomp.c)

עליכם לכתוב תוכנית מחשב אינטראקטיבית הקוראת פקודות מהקלט הסטנדרטי, מפענחת ומבצעת אותן. הפקודות עוסקות בפעולות אריתמטיות על מספרים מרוכבים.

## תזכורת מספרים מרוכבים:

מספר מרוכב (complex) הוא מספר בן שני חלקים: חלק ממשי וחלק מדומה, כאשר ביניהם רשום הסימן "+" או הסימן "-".

מבנה המספר הוא:  $a + bi$  כאשר  $a$  החלק הממשי ו- $bi$  החלק המדומה. החלק המדומה הוא מכפלה של שני גורמים:  $b$  הוא מספר ממשי, ואילו  $i$  מציין את השורש הריבועי של המספר -1,

$$i = \sqrt{-1}$$

דוגמאות של מספרים מרוכבים:

$$24.65 + (15.376)i \quad 87.5 - (14.3)i \quad -153 + 24i$$

להלן הגדרות של הפעולות החשבוניות הבסיסיות על מספרים מרוכבים:

חיבור של שני מספרים מרוכבים:

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

חסור של שני מספרים מרוכבים:

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

כפל של מספר מרוכב במספר ממשי:

$$m * (a + bi) = ma + (mb)i$$

כפל של מספר מרוכב במספר מדומה:

$$mi * (a + bi) = mi * a + mi * bi = -mb + (ma)i$$

כפל של מספר מרוכב במספר מרוכב:

$$(a + bi) * (c + di) = a * c + a * di + bi * c + bi * di = (ac - bd) + (ad + bc)i$$

חישוב הערך המוחלט של מספר מרוכב (התוצאה היא מספר ממשי אי-שלילי):

$$|a + bi| = \sqrt{a^2 + b^2}$$

משימות התוכנית:

עליכם להגדיר, באמצעות שימוש ב-`typedef` את הטיפוס `complex` אשר מחזיק מספר מרוכב. על מבנה הנתונים שבחרתם להיות יעיל מבחינת כמות זיכרון הנדרשת, ויעיל מבחינת הגישה אליו.

בנוסף, עליכם להגדיר בתוכנית הראשית 6 משתנים: `A,B,C,D,E,F` מטיפוס `complex`.

בתחילת הריצה, יש לאתחל את כל ששת המשתנים לאפס (הערך המרוכב  $0+0i$ ).

כעת, עליכם לבצע פעולות חשבוניות עם מספרים מרוכבים. כל פעולה תופעל באמצעות פקודה שמועברת בקלט לתוכנית, כמפורט להלן. בפקודות אלה, אופרנד שהוא משתנה מרוכב יהיה אחד מששת המשתנים שהוגדרו לעיל.

#### מפרט הפקודות המשמשות כקלטים לתוכנית:

1. הצבת מספר מרוכב במשתנה:

**מספר-ממשי-שני, מספר-ממשי-ראשון, שם-משתנה-מרוכב read\_comp**

הפקודה תגרום להצבת ערך מרוכב במשתנה המרוכב ששמו מופיע בפקודה. המספר הממשי הראשון הוא החלק הממשי של המספר המרוכב, והמספר הממשי השני הוא החלק המדומה של המספר המרוכב (החלק המדומה נתון בפקודה ללא הגורם  $i$ )

לדוגמה, הפקודה הבאה:

`read_comp A, 5.1, 6.23`

תבצע את ההצבה:

$$A = 5.1 + (6.23)i$$

הערה: זוהי הפקודה היחידה שמשנה את ערכו של משתנה מרוכב בתוכנית.

2. הדפסת ערך של משתנה מרוכב:

**שם-משתנה-מרוכב print\_comp**

ערכו של המשתנה המרוכב ששמו ניתן בפקודה יודפס בצורה נאה בפלט.

לדוגמה, הפקודה הבאה:

`print_comp A`

תגרום להדפסת ערך המשתנה  $A$ . בהנחה שהפקודה היא בהמשך לדוגמה בסעיף 1, יודפס:

$$5.10 + (6.23)i$$

הערה: יש להדפיס כל מספר עם דיוק של לפחות שתי ספרות מימין לנקודה.

3. חיבור מספרים מרוכבים:

**שם-משתנה-מרוכב-ב', שם-משתנה-מרוכב-א' add\_comp**

יתבצע חיבור של שני המספרים המרוכבים אשר במשתנים המופיעים בפקודה:

מספר-מרוכב-ב' + מספר-מרוכב-א'

תוצאת הפעולה תודפס לפלט בפורמט דומה לפורמט ההדפסה של סעיף 2.

4. חיסור מספרים מרוכבים:

**שם-משתנה-מרוכב-ב', שם-משתנה-מרוכב-א' sub\_comp**

יתבצע חיבור של המספר המרוכב במשתנה ב' מן המספר המרוכב במשתנה א':

מספר-מרוכב-ב' – מספר-מרוכב-א'

תוצאת הפעולה תודפס לפלט בפורמט דומה לפורמט ההדפסה של סעיף 2.

5. כפל מספר מרוכב עם מספר ממשי:

**מספר-ממשי, שם-משתנה-מרוכב mult\_comp\_real**

יתבצע כפל של המשתנה המרוכב והמספר הממשי הנתונים בפקודה.

מספר-ממשי \* מספר-מרוכב

תוצאת הפעולה תודפס לפלט בפורמט דומה לפורמט ההדפסה של סעיף 2.

6. כפל מספר מרוכב עם מספר מדומה:

**מספר-מדומה, שם-משתנה-מרוכב mult\_comp\_img**

יתבצע כפל של המשתנה המרוכב והמספר המדומה הנתונים בפקודה.

המספר המדומה נתון בפקודה ללא הגורם i.

(i \* מספר-מדומה) \* מספר-מרוכב

תוצאת הפעולה תודפס לפלט בפורמט דומה לפורמט ההדפסה של סעיף 2.

7. כפל שני מספרים מרוכבים:

**שם-משתנה-מרוכב-ב', שם-משתנה-מרוכב-א' mult\_comp\_comp**

יתבצע כפל של שני המשתנים המרוכבים המופיעים בפקודה:

מספר-מרוכב-ב' \* מספר-מרוכב-א'

תוצאת הפעולה תודפס לפלט בפורמט דומה לפורמט ההדפסה של סעיף 2.

8. ערך מוחלט של מספר מרוכב:

**שם-משתנה-מרוכב abs\_comp**

יחושב ערכו המוחלט של המשתנה המרוכב שמופיע בפקודה:

| מספר-מרוכב |

תוצאת הפעולה תודפס לפלט בפורמט דומה לפורמט ההדפסה של סעיף 2.

9. סיום התוכנית:

**stop**

פקודה זו היא ללא פרמטרים, ומטרתה לסיים את התוכנית.

### המבנה התחבירי של הקלט:

- כל פקודה תופיע בשלמותה בשורת קלט יחידה, כולל כל הפרמטרים. מותרות גם שורות ריקות (שורות המכילות רק תווים לבנים).
- שם הפקודה מופרד מהפרמטר הראשון באמצעות רווחים ו/או טאבים (אחד או יותר).
- בין כל שני אופרנדים 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 לפני שם הפקודה, וגם בסוף השורה (אחרי הפרמטר האחרון).
- אסור שיהיו תווי זבל בסוף השורה (למעט תווים לבנים).
- שמות הפקודות יופיעו באותיות קטנות בלבד, ושמות המשתנים באותיות גדולות בלבד.

### אופן פעולת התוכנית:

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

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

יש לטפל גם במצב של EOF (גמר הקלט). עצירת התוכנית שלא באמצעות פקודת stop אינה נחשבת תקינה (גם לא כאשר הקלט מגיע מקובץ באמצעות redirection), ויש להדפיס הודעת שגיאה על כך ורק אז לעצור. שימו לב: השורה האחרונה בקובץ קלט אינה חייבת להסתיים בתו 'n'. במקרה כזה, אם יש בשורה האחרונה פקודה, יש לטפל בה כרגיל (סוף הקובץ מסמן את סוף הפקודה).

### 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\_comp G, 3.1, 6.5  
יש להגיב בהודעה כגון: Undefined complex variable
2. לפקודה: read\_comp a, 3.6, 5.1  
יש להגיב בהודעה כגון: Undefined complex variable
3. לפקודה: do\_it A, B  
יש להגיב בהודעה כגון: Undefined command name
4. לפקודה: Add\_Comp A, C  
יש להגיב בהודעה כגון: Undefined command name
5. לפקודה: read\_comp A, 3.5, xyz  
יש להגיב בהודעה כגון: Invalid parameter – not a number
6. לפקודה: read\_comp A, 3.5

Missing parameter	יש להגיב בהודעה כגון:
read_comp A, 3.5, -3,	7. לפקודה:
Extraneous text after end of command	יש להגיב בהודעה כגון:
add_comp B	8. לפקודה:
Missing parameter	יש להגיב בהודעה כגון:
print_comp C, D	9. לפקודה:
Extraneous text after end of command	יש להגיב בהודעה כגון:
sub_comp F, , D	10. לפקודה:
Multiple consecutive commas	יש להגיב בהודעה כגון:
mult_comp_comp F D	11. לפקודה:
Missing comma	יש להגיב בהודעה כגון:
mult_comp_real, A, 2.5	12. לפקודה:
Illegal comma	יש להגיב בהודעה כגון:
mult_comp_img A, B	13. לפקודה:
Invalid parameter – not a number	יש להגיב בהודעה כגון:
abs_comp	14. לפקודה:
Missing parameter	יש להגיב בהודעה כגון:
abs_comp 2.5	15. לפקודה:
Undefined complex variable	יש להגיב בהודעה כגון:
stop A	16. לפקודה:
Extraneous text after end of command	יש להגיב בהודעה כגון:

#### להלן דוגמה של סדרת פקודות שכולן תקינות:

הערה: סדרה כגון זו יכולה לשמש כקלט בהרצת בדיקה של התוכנית על קלט תקין.

```

print_comp A
print_comp B
print_comp C
read_comp A, 45.1, -23.75
print_comp A
read_comp B, 54.2, 3.56
print_comp B
read_comp C, 0, -1

```

```

print_comp C
add_comp A, B
sub_comp C, A
sub_comp B, B
sub_comp D, A
mult_comp_real A, 2.51
mult_comp_img A, -2.564
mult_comp_comp A, B
mult_comp_comp E, C
abs_comp A
abs_comp B
abs_comp C
abs_comp F
stop

```

### דרישות והנחיות נוספות:

- יש לחלק את התוכנית למספר קבצי מקור: `complex.c`, `mycomp.c` ו-`complex.h`.
  - בקובץ `mycomp.c` תהיה התוכנית הראשית `main`, וכן כל פעילויות האינטראקציה עם המשתמש וניתוח הקלט (לרבות הדפסת הודעות השגיאה). כמו כן, יוגדרו בקובץ זה ששת המשתנים מטיפוס `complex`.
  - בקובץ `complex.c` יש לרכז את הפעולות החשבוניות על מספרים מרוכבים. לכל פעולה יש לממש פונקציה נפרדת, עם פרמטרים לפי מפרט הפעולה המוגדר לעיל. אין לבצע ניתוח של הקלט או הדפסות מתוך קובץ זה, למעט הדפסת המספר המרוכב כנדרש בפעולה `print_comp`.
  - בקובץ `complex.h` תהיה הגדרת טיפוס הנתונים `complex`, וכן ההצהרות (אב-טיפוס) של הפונקציות הממומשות בקובץ `complex.c`. יש לכלול (`#include`) את הקובץ `complex.h` בקבצי המקור האחרים.
  - באפשרותכם לבנות קבצי מקור נוספים (למשל: קובץ המכיל פונקציות עזר לניתוח הקלט, וכד').
- הקלט לתוכנית הוא מ-`stdin`, ויכול להגיע **מהמקלדת או מקובץ** (באמצעות `redirection` בעת הפעלת התוכנית). לנוחיותכם, הכינו מספר קבצי קלט והשתמשו בהם שוב ושוב לדיבוג התוכנית. בכל קובץ קלט תהיה סדרה של פקודות על מספרים מרוכבים.
- לפני הניתוח של כל שורת קלט, על התוכנית **להדפיס באופן יזום את השורה לפלט, בדיוק כפי שנקראה**. זאת כדי שניתן יהיה לראות בפלט את הפקודות המקוריות, גם כאשר הקלט מגיע מקובץ.

**חובה לצרף להגשה הרצות בדיקה (אחת או יותר), המדגימות את השימוש בכל סוגי הפקודות ובכל ששת המשתנים המרוכבים, וכן את הטיפול בכל מגוון השגיאות בקלט.**  
**רמז:** מומלץ להכניס בקלט פקודת הדפסה של התוצאה אחרי כל פעולה, כדי להראות שהתוצאה אכן נכונה (ראו לעיל הדוגמה של סדרת פקודות תקינות).  
**יש להגיש תדפיס מסך (וקובץ פלט) של כל ההרצות.** אם תשתמשו בקבצי קלט, יש להגיש גם קבצים אלה.

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

**בהצלחה!**