

תרגיל בית מספר 1

נושא: שפת C++ כשפת C משופרת

תאריך הגשה: יום ג', 04/04/2023, 23:59

הגשה ביחידים

בהצלחה רבה!

תיאור התרגיל

מטרתו של תרגיל בית זה הינה כפולה – הראשונה, יישור קו מבחינת רמת התכנות הנדרשת בתחילת הקורס; השנייה, תכנות בשפת C++ כשפת C משופרת, אך לאו דווקא עם דגש על השפה כמונחית עצמים בשלב זה. התרגיל יעסוק בבניית מחשבון אלגברי מעל המרוכבים עבור מטריצות ריבועיות, אשר ישלב רכיבי קלט/פלט, עיבוד טקסטואלי, וניהול זיכרון. להנחיות מצורף נספח המספק את ההגדרות המתמטיות הנדרשות לפתרון התרגיל.

מחשבון מרוכב: אפיון

עליכם לתכנן ולממש מחשבון התומך בפעולות חישוב מתקדמות עבור מטריצות ריבועיות מעל שדה המספרים המרוכבים:

- [0] יציאה מן המחשבון
- [1] שמירת מטריצה ממימד כלשהו בזיכרון המחשבון תחת שם נתון
- [2] מחיקת מטריצה עם שם נתון מזיכרון המחשבון
- [3] חישוב מכפלה של מטריצה בסקלר מרוכב
- [4] העלאת מטריצה בחזקה אי-שלילית
- [5] חישוב דטרמיננטה של מטריצה (ראו נספח מתמטי)
- [6] הדפסת מטריצה אל ערוץ הפלט הסטנדרטי

עם הרצת התכנית, המערכת תמתין לבחירת המשתמש בהתאם לתפריט הפעולות שהוצג (**אין להדפיס את התפריט**); לאחר בחירה וביצוע פעולות [1] עד [6] המערכת תחזור למצב "המתנה" לבחירה הבאה, ורק פעולה [0] תביא לסיומה של התכנית. התקשורת מול המשתמש תתנהל בערוצי הקלט והפלט הסטנדרטיים, כאשר הודעות שגיאה תצאנה אל ערוץ פלט השגיאה. במקרה שקלט המשתמש הוא מספר פעולה שאינו קיים (שונה מ-[0-6]), יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid command; type 0 for exit
```

שמירת מטריצה בזיכרון המחשבון

בבחירת פעולה זו [1], המערכת תקלוט מן המשתמש מטריצה ותשמור אותה במבנה הנתונים הפנימי שלה תחת שם שיימסר ע"י המשתמש. קליטת מטריצה מן המשתמש תתנהל לפי הפרוטוקול הבא (ללא טקסט הנחייה):

א. הקלדת שם המטריצה (זהו המזהה הייחודי של כל מטריצה, המהווה מחרוזת בעלת אותיות קטנות בלבד [a-z]) והקלדת Enter.

ב. הזנת מימד המטריצה המיועדת (מספר חיובי) והקלדת Enter.

ג. ערכי שורות המטריצה (מספרים מרוכבים), כאשר כל ערך מרוכב יוגדר באמצעות צמד ממשיים בסוגריים וביניהם פסיק (לציון רכיבים ממשי ומדומה), וכל ערך יופרד מערך אחר באמצעות רווח; הקלדת Enter בין שורה לשורה.

ד. עדכון מבנה הנתונים הפנימי וחזרה להמתנה בתפריט הראשי

במקרה שמטריצה בעלת שם זהה כבר קיימת בזיכרון המחשבון, עדכון מאוחר ידרוס לחלוטין כל מידע קודם שהוזן למחשבון. במקרה שקלט המשתמש איננו חוקי (פורמט שם, מימד שאינו חיובי, קלט שגוי עבור שורות המטריצה), יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid input
```

תכנות בשפת ++C, אביב 2023

מחיקת מטריצה מזיכרון המחשבון

בבחירת פעולה זו [2], על המחשבון למחוק מהזיכרון הפנימי מטריצה יחידה ששמה נמסר ע"י המשתמש. הפרוטוקול כולל את הקלדת שם המטריצה למחיקה ולאחר מכן Enter. בהינתן שם חוקי אשר קיים בזיכרון, תימחק המטריצה ולא יודפס פלט.

בהינתן קלט שאינו חוקי (שם מטריצה עם תווים שאינם ב- [a-z]), יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid input
```

בהינתן שם חוקי שאינו קיים בזיכרון, יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: <name> does not exist
```

חישוב מכפלת מטריצה בסקלר

בבחירת פעולה זו [3], המחשבון יכפול מטריצה שקיימת בזיכרון בסקלר שיימסר ע"י המשתמש – וישמור את התוצאה בזיכרון. קליטת הנתונים תתנהל לפי הפרוטוקול הבא (ללא טקסט הנחייה):

א. שם המטריצה לכפל בסקלר והקלדת Enter

ב. סקלר מרוכב (מיוצג באמצעות צמד ממשיים בסוגריים, כאשר ביניהם פסיק) והקלדת Enter

ג. שם המטריצה שתהווה את תוצאת החישוב ותישמר בזיכרון המחשבון; שם המטריצה יכול להיות זהה לשם המטריצה שנכפלה בסקלר, ובמקרה זה, המטריצה פשוט תעודכן בתוצאת החישוב.

בהינתן נתונים חוקיים, יבוצע החישוב והתוצאה תעודכן בזיכרון הפנימי; המחשבון יחזור להמתנה בתפריט הראשי. בהינתן קלט שאינו חוקי (פורמט שגוי עבור הסקלר או שם מטריצת התוצאה) יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid input
```

בהינתן שם מטריצה חוקי שאינו קיים בזיכרון, יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: <name> does not exist
```

העלאת מטריצה בחזקה

בבחירת פעולה זו [4], המחשבון יעלה מטריצה שקיימת בזיכרון בחזקה שלמה אי-שלילית שתימסר ע"י המשתמש וישמור את התוצאה בזיכרון, כלומר, יחשב את תוצאת המכפלה של המטריצה בעצמה כמספר הפעמים שנמסר ע"י המשתמש. בהינתן חזקת האפס, התוצאה מוגדרת כמטריצת היחידה. קליטת הנתונים תתנהל לפי הפרוטוקול הבא (ללא טקסט הנחייה):

א. שם המטריצה להעלאת בחזקה והקלדת Enter

ב. מספר שלם (אי-שלילי) והקלדת Enter

ג. שם המטריצה שתהווה את תוצאת החישוב ותישמר בזיכרון המחשבון; שם המטריצה יכול להיות זהה לשם המטריצה שהועלתה בחזקה, ובמקרה זה, המטריצה פשוט תעודכן בתוצאת החישוב.

בהינתן נתונים חוקיים, יבוצע החישוב והתוצאה תעודכן בזיכרון הפנימי; המחשבון יחזור להמתנה בתפריט הראשי. בהינתן קלט שאינו חוקי (חזקה בלתי-חוקית או פורמט שגוי עבור שם מטריצת התוצאה) יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid input
```

בהינתן שם מטריצה חוקי שאינו קיים בזיכרון, יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: <name> does not exist
```

חישוב דטרמיננטה של מטריצה

בבחירת פעולה זו [5], המחשבון יבצע חישוב דטרמיננטה של מטריצה הקיימת בזיכרון; תוצאת החישוב הינה סקלר מרוכב. החישוב המתמטי מוגדר היטב בנספח, ואתם נדרשים לממש אותו ברוסיה. הפרוטוקול כולל את הקלדת שם המטריצה לחישוב ולאחר מכן Enter. בהינתן שם חוקי אשר קיים בזיכרון, יבוצע החישוב ותוצאתו תודפס לפלט הסטנדרטי. בהינתן קלט שאינו חוקי (שם מטריצה עם תווים שאינם ב- [a-z]), יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid input
```

בהינתן שם מטריצה חוקי שאינו קיים בזיכרון, יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: <name> does not exist
```

שימו לב שפעולת חישוב זו, בניגוד לקודמותיה, מחזירה סקלר מרוכב אשר אינו נשמר בזיכרון המחשבון, אלא רק מודפס לערוץ הפלט הסטנדרטי בתום החישוב.

תכנות בשפת ++C, אביב 2023

הדפסת מטריצה

בבחירת פעולה זו [6], המערכת תדפיס לערוץ הפלט הסטנדרטי מטריצה מסוימת ששמה יימסר ע"י המשתמש. הפרוטוקול כולל את הקלדת שם המטריצה להדפסה ולאחר מכן Enter. בהינתן שם חוקי אשר קיים בזיכרון, תודפס המטריצה לפלט הסטנדרטי. בהינתן קלט שאינו חוקי (שם מטריצה עם תווים שאינם ב-[a-z]), יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: invalid input
```

בהינתן שם שאינו קיים בזיכרון, יש להוציא את ההודעה הבאה ולאחר מכן לחזור לתפריט הראשי:

```
error: <name> does not exist
```

הנחות עבודה ופורמטים

- על מבנה הנתונים בו מאוחסנות המטריצות להיות מוקצה דינאמית על ה-Heap. יש לכם חופש מוחלט בבחירת טיפוס מבנה הנתונים.
- שם מטריצה הינו מחרוזת ייחודית שאינה עולה באורכה על 4 תווים קטנים [a-z].
- ניתן להניח כי מימד המטריצות המקסימלי הינו 256.
- מספר המטריצות שניתן לאחסן בזיכרון אינו מוגבל בעיקרון (אך בפועל יש 456976 מחרוזות ייחודיות לשמות חוקיים).

פורמט הדפסה

על כל אלמנט מספרי להיות מודפס עם **לכל היותר שלוש ספרות אחרי הנקודה**; אין להדפיס נקודה עשרונית אם אין צורך בכך.

על מספר מרוכב להיות מיוצג באמצעות צמד מספרים ממשיים בתוך סוגריים, מופרדים בפסיק יחיד. להלן שלוש דוגמאות עבור שלושה מספרים מרוכבים שונים:

(1,0) (0,1) (0.707,0.707)

הדפסת מטריצה (פעולה [6]): השורה הראשונה תכיל את שם המטריצה, סימן שווה, וסוגר מרובע שמאלי – מופרדים באמצעות רווח. כל אחת מן השורות הבאות תכיל שורת אלמנטים של המטריצה, בפורמט שהוגדר עבור אלמנט מרוכב, כאשר כל אלמנט יופרד מאחר באמצעות רווח בודד. השורה האחרונה בהדפסה, שתבוא לאחר השורה האחרונה של המטריצה, תכיל רק סוגר מרובע ימני. לדוגמא, מטריצה ממימד 5 בעלת השם a , תודפס כך לפלט הסטנדרטי:

$a = [$

(1,0) (0,1) (0,0) (0,0) (0.5,0.5)

(0,-1) (1,0) (0,0) (0,0) (0,0)

(0,0) (0,0) (1,0) (0,0) (0,0.256)

(0,0) (0,0) (0,0) (1,0) (0,0)

(0.5,-0.5) (0,0) (0,-0.256) (0,0) (1,0)

]

תכנות בשפת C++, אביב 2023

דגשים

- יש לתכנן מראש את מבנה התכנית, ולהגדיר בהתאם את המרכיבים איתם תעבדו.
- יש לממש את עבודת הקלט והפלט באמצעות רכיבי C++ ולא באמצעות פונקציות C.
- הקצאה דינאמית צריכה להתבצע רק באמצעות הפקודות new ו-delete.
- עבודה עם מחרוזות יכולה להתבצע באמצעות טיפוס `std::string`, אך אין דרישה לכך.
- בתרגיל בית זה אין להשתמש בספריית STL; מבני הנתונים והאלגוריתמים צריכים להיות ממומשים על-ידיכם.**
בפרט, את פעולת חישוב הדטרמיננטה עליכם לממש מפורשות באמצעות רקורסיה.
- יש לבדוק תקינות קלטים ולהציג הודעות שגיאה מתאימות.
- עליכם לוודא כי התכנית עוברת קומפילציית g++ התואמת את הקומפיילר שעל שרת החוג ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המודולים והפונקציות השונים.
- יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות של ההגשה: `hwcheck`.

הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('48'), שם התרגיל ('ex1') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי בפורמט הבא: `48_ex1_studID.zip`.
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות `*.cpp` `*.h` `*.txt`.
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם `48_ex1_012345678.zip` הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.
לא תתקבלנה הגשות באיחור!

בנספח זה ניתנת ההגדרה המתמטית הנדרשת למימוש חישוב הדטרמיננטה. בהינתן מטריצה ריבועית מרוכבת A מממד n ,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \in \mathbb{C}^{n \times n}$$

הדטרמיננטה שלה, אשר מסומנת ע"י $\det(A)$ או $|A|$, הינה סקלר מרוכב המוגדר באמצעות פעולת חישוב רקורסיבית.

דטרמיננטה של מטריצה ריבועית מממד 2 מוגדרת להיות הסקלר הבא:

$$\det \begin{pmatrix} a & b \\ c & d \end{pmatrix} \equiv \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

עבור מימד n כלשהו, ביטוי הדטרמיננטה מוגדר באמצעות דטרמיננטות מסדר $n-1$ אשר עוברות סכימה עם סימנים מתחלפים:

$$\det \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = a_{11} \cdot \det \begin{pmatrix} a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n2} & \dots & a_{nn} \end{pmatrix} - a_{12} \cdot \det \begin{pmatrix} a_{21} & a_{23} & \dots & a_{2n} \\ \vdots & \ddots & & \vdots \\ a_{n1} & a_{n3} & \dots & a_{nn} \end{pmatrix} + \dots \pm a_{1n} \cdot \det \begin{pmatrix} a_{21} & a_{22} & \dots & a_{2(n-1)} \\ \vdots & \ddots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n(n-1)} \end{pmatrix}$$

לפיכך, הנוסחה הכללית נתונה באמצעות:

$$\det(A) = \sum_{j=1}^n (-1)^{j+k} a_{jk} \mathbf{M}_{jk}$$

כאשר \mathbf{M}_{jk} נגזר מתוך A ע"י מחיקת השורה ה- j והעמודה ה- k .