

מבוא למדעי המחשב - סמסטר ב' תשע"ט

עבודת בית מספר 1

צוות התרגיל: דניאל אלדמ

תאריך פרסום: 6.3.19

תאריך הגשה: 21.3.19, 12:00 בצהריים

הוראות מקדימות:

הגשת עבודות בית

1. קראו את העבודה מתחילתה ועד סופה לפני שאתם מתחילים לפתור אותה. ודאו שאתם מבינים את כל המשימות. רמת הקושי של המשימות אינה אחידה: הפתרון של חלק מהמשימות קל יותר, ואחרות מצריכות חקירה מתמטית - שאותה תוכלו לבצע בספרייה או בעזרת מקורות דרך רשת האינטרנט. בתשובות שבהן אתם מסתמכים על עובדות מתמטיות שלא הוצגו בשיעורים, יש להוסיף כהערה במקום המתאים בקוד את ציטוט העובדה המתמטית ואת המקור (כגון ספר או אתר).
2. עבודה זו תוגש בזוגות. הגשה שלא בזוגות תגרור הורדה בציון. על מנת להגיש את העבודה יש להירשם למערכת ההגשות (Submission System). את הרישום למערכת ההגשות מומלץ לבצע כבר עכשיו, טרם הגשת העבודה (קחו בחשבון כי הגשה באיחור לא מתקבלת). את הגשת העבודה ניתן לבצע רק לאחר הרישום למערכת. שימו לב שיש לפתוח קבוצת הגשה אחת עבור כל זוג – על סטודנט אחד לפתוח את הקבוצה ולצרף את הסטודנט השני לעבודה מצורפים קובצי Java עם שמות כגון Task<n>.java, כאשר <n> מציינ את מספר המשימה המתאימה לקובץ (לדוגמא, קובץ Java בשם Task2.java מתאים למשימה מספר 2). צרו תיקייה חדשה והעתיקו את קבצי ה-Java לתוכה. עליכם לערוך את הקבצים האלו בהתאם למפורט בתרגיל ולהגישם כפתרון, מכווצים כקובץ ZIP יחיד. שימו לב: עליכם להגיש רק את קבצי ה-Java. אין לשנות את שמות הקבצים. אין להגיש קבצים נוספים. שם קובץ ה-ZIP יכול להיות כרצונכם, אך באנגלית בלבד. בנוסף, הקבצים שתגישו יכולים להכיל טקסט המורכב מאותיות באנגלית, מספרים וסימני פיסוק בלבד. טקסט אשר יכיל תווים אחרים (אותיות בעברית, יוונית וכד'..) לא יתקבל.
4. קבצים שיוגשו שלא על פי הנחיות אלו לא ייבדקו. את קובץ ה-ZIP יש להגיש ב-Submission System. פרטים בעניין ההרשמה ואיך להגיש את העבודה תוכלו למצוא באתר.

בדיקת עבודות הבית

5. עבודות הבית נבדקות גם באופן ידני וגם באופן אוטומטי. הבדיקה האוטומטית מתייחסת לפלט התכנית המודפס למסך. לכן, יש להקפיד על ההוראות ולבצע אותן במדויק. כל הדפסה אשר אינה עונה בדיוק על הדרישות המופיעות בעבודה (כולל שורות, רווחים, סימני פיסוק או כל תו אחר - מיותרים, חסרים או מופיעים בסדר שונה מהנדרש), לא תעבור את הבדיקה האוטומטית ולכן תגרור פגיעה בציון.
6. סגנון כתיבת הקוד ייבדק באופן ידני. יש להקפיד על כתיבת קוד ברור, על מתן שמות משמעותיים למשתנים, על הזחות (אינדנטציה), ועל הוספת הערות בקוד המסבירות את תפקידם של מקטעי הקוד השונים. אין צורך למלא את הקוד בהערות סתמיות, אך חשוב לכתוב הערות בנקודות קריטיות, המסבירות קטעים חשובים בקוד. הערות יש לרשום אך ורק באנגלית. כתיבת קוד אשר אינה עומדת בדרישות אלו תגרור הפחתה בציון העבודה.
7. בכל פעם שאתם מתבקשים להדפיס למסך, עליכם להשתמש בפונקציה System.out.println(), אשר מדפיסה למסך ויורדת שורה (לכן, כל פעולת הדפסה תופיע בשורה נפרדת). אין להדפיס למסך דברים מיותרים (כגון: "please enter an integer").
8. בכדי לקלוט נתונים מהמשתמש יש להשתמש ב-Scanner, כפי שנלמד בכיתה.

עזרה והנחיה

9. לכל עבודת בית בקורס יש צוות שאחראי לה. ניתן לפנות לצוות בשעות הקבלה. פירוט שמות האחראים לעבודה מופיע במסמך זה וכן באתר הקורס, כמו גם פירוט שעות הקבלה. בשאלות טכניות אפשר גם לגשת לשעות התגבורים בהם ניתנת עזרה במעבדה. כמו כן, אתם יכולים להיעזר בפורום ולפנות בשאלות לחבריכם לכיתה. צוות הקורס עובר על השאלות ונותן מענה במקרה הצורך.
10. בכל בעיה אישית הקשורה בעבודה (מילואים, אשפוז וכו'), אנא פנו אלינו דרך מערכת הפניות, כפי שמוסבר באתר הקורס.

הערות ספציפיות לעבודת בית זו

11. בעבודה זו 7 משימות וסך הנקודות המקסימלי הוא 105. כלומר, יש אפשרות לקבל 5 נקודות בonus. שימו לב שמספר הנקודות לכל משימה אחיד (15 נקודות למשימה) ואינו מצביע על קושי המשימה.
12. בעבודה זו מותר להשתמש בידע שנלמד עד הרצאה 3 (כולל), וכן עד תרגול 2 (כולל). לא ניתן להשתמש במערכים, מחרוזות, פונקציות, או כל צורת קוד אחרת אשר לא נלמדה בכיתה.
13. בעבודה זו, כל המשתנים עבור מספרים שלמים צריכים להיות מטיפוס `int`.

יושר אקדמי

הימנעו מהעתקות! ההגשה היא ביחידים. אם מוגשות שתי עבודות עם קוד זהה או אפילו דומה - זוהי העתקה, אשר תדווח לאלתר לוועדת משמעת. אם טרם עיינתם ב**סילבוס הקורס** אנא עשו זאת כעת.

מומלץ לקרוא היטב את כל ההוראות המקדימות ורק לאחר מכן להתחיל בפתרון המשימות. ודאו שאתם יודעים לפתוח קבוצת הגשה במערכת ההגשות.

משימת חימום

משימה 1 - משימת חימום (15 נקודות)

פתחו את הקובץ Task1.java וכתבו בו תכנית אשר מקבלת מהמשתמש שלושה מספרים שלמים ומדפיסה אותם למסך בסדר עולה (זוהי פעולת מיון).

דוגמאות:

אם בקלט שלושת הערכים הם 3, 5 ו-1, אזי הפלט יהיה:

1

3

5

דוגמה נוספת: אם בקלט שלושת הערכים הם 3, -5 ו-3, אזי הפלט יהיה:

-5

3

3

להזכירכם, **אין** להדפיס דברים מיותרים למסך (כגון: "please enter three integers").

פירוק לגורמים ראשוניים

תזכורת: מספר ראשוני הינו מספר שלם הגדול ממש מ-1 אשר מתחלק רק ב-1 ובעצמו.

דוגמאות: $2, 3, 5, 7, 11, 13, 17, \dots$ (ישנם אינסוף מספרים ראשוניים).

עובדה ידועה בשם Fundamental theorem of arithmetic/ Unique factorization theorem מציינת כי כל מספר שלם הגדול מ-1 הינו מספר ראשוני או ניתן להצגה כמכפלה של מספרים ראשוניים (הנקראים גם גורמים ראשוניים). מכפלה זו יחידה עד כדי סדר המספרים.

דוגמאות: $100 = 2 * 2 * 5 * 5$, $6 = 2 * 3$, $23 = 23$

משימה 2 – הדפסת הגורמים הראשוניים של מספר נתון (15 נקודות)

פתחו את הקובץ Task2.java וכתבו בו תכנית אשר מקבלת מהמשתמש מספר שלם חיובי גדול ממש מ-1 ומדפיסה למסך את כל הגורמים הראשוניים שלו (כולל כפילויות) בסדר לא יורד. ניתן להניח כי הקלט תקין. כלומר, שהקלט הוא מספר שלם וגדול מ-1.

להזכרכם, אין להדפיס דברים מיותרים למסך (כגון: "please enter an integer").

דוגמאות:

אם הקלט הוא 100, הפלט יהיה:

2

2

5

5

אם הקלט הוא 6, הפלט יהיה:

2

3

אם הקלט הוא 23, הפלט יהיה:

23

סיכום שברים

משימה 3 – סיכום שני שברים (15 נקודות)

פתחו את הקובץ Task3.java וכתבו בו תכנית אשר מקבלת מהמשתמש ארבעה מספרים שלמים וחיוביים a, b, c, d (בסדר זה) ומדפיסה זוג מספרים שלמים, x ו- y , כך ש-

$$\frac{a}{b} + \frac{c}{d} = \frac{x}{y}$$

התוכנית תדפיס את x ואחריו בשורה נפרדת את y . המספרים x ו- y ייצגו שבר מצומצם (אין להם גורם משותף גדול מ-1). ניתן להניח כי הקלט a, b, c, d חוקי, כלומר מספרים שלמים וחיוביים.

דוגמאות:

אם הקלט הוא המספרים 1,2,1,3, הפלט יהיה

5

6

$$\text{כיוון ש-} \frac{1}{2} + \frac{1}{3} = \frac{5}{6}$$

אם הקלט הוא המספרים 8,4,7,5, הפלט יהיה

17

5

$$\text{כיוון ש-} \frac{8}{4} + \frac{7}{5} = \frac{17}{5}$$

משימה 4 – סיכום מספר כלשהו של שברים (15 נקודות)

פתחו את הקובץ Task4.java וכתבו בו תכנית אשר פותרת את המשימה הקודמת, בהבדל אחד: במקום להניח כי הקלט הוא בדיוק שני זוגות של מספרים שלמים וחיוביים, כעת הוא יהיה מספר שלם וחיובי כלשהו של זוגות מספרים שלמים וחיוביים (לפחות זוג אחד). בתום הזנת זוגות המספרים, המשתמש יורה לכם לעצור ע"י הכנסת הקלט: 1- (המספר מינוס אחד).

דוגמאות:

1. אם הקלט הוא המספרים: 1,2,1,3,1,4,-1 (משמאל לימין), הפלט יהיה

13

12

$$\text{כיוון ש-} \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{13}{12}$$

2. אם הקלט הוא: 8,4,-1, הפלט יהיה

2

1

$$\text{כיוון ש-} \frac{8}{4} = \frac{2}{1}$$

בחירה על ידי שיחלוף-מותנה

בשלוש המשימות הבאות נעסוק באלגוריתמי בחירה המוגדרים במונחים של פעולה הנקראת "שיחלוף-מותנה". שיחלוף-מותנה של שני ערכים מספריים x ו- y היא פעולה שמחליפה בין הערכים של x ו- y , במידת הצורך כדי להבטיח ש- $x \leq y$ וניתן לממש אותה בג'אווה (בהנחה שהמשתנים x ו- y הוגדרו להיות מסוג `int`) בעזרת קטע הקוד הבא:

```
if (x>y) {  
    int tmp = y;  
    y = x;  
    x = tmp;  
}
```

במשפט ה-`if` הנ"ל הקוד בבלוק משחלף את הערכים במקרה שהתנאי מתקיים. במשימות 5 ו-6 נתייחס גם למספר פעולות השיחלוף-המותנה שבתכנית.

נציין כי פתרון נכון של משימות 5 ו-6 ייחשב לפתרון טוב יותר ככל שיהיו בו פחות פעולות שיחלוף-מותנה.

משימה 5 – מציאת שני מספרים קטנים 1 (15 נקודות)

פתחו את הקובץ `Task5.java` וכתבו בו תכנית אשר מקבלת מהמשתמש שלושה מספרים שלמים, a , b ו- c . התוכנית תדפיס שני מספרים קטנים ביותר מתוך השלושה, בסדר כלשהו.

דוגמאות:

אם הקלט הוא המספרים $a = 2, b = 1, c = 3$ התכנית יכולה להדפיס

2

1

או

1

2

אם הקלט הוא המספרים $a = 2, b = 1, c = 1$ התכנית תדפיס

1

1

אתם מתבקשים לתת פתרון מבוסס פעולות של שיחלוף-מותנה. שימו לב: בקוד שתכתבו, אין להשתמש במשפטי `if` או `if-else` שלא בהקשר של פעולת שיחלוף-מותנה. עובדה מתמטית: ניתן לפתור משימה זו עם שתי פעולות שיחלוף-מותנה, אך לא עם פחות מכך.

משימה 6 - מציאת שני מספרים קטנים 2 (15 נקודות)

פתחו את הקובץ Task6.java וכתבו בו תכנית אשר מקבלת מהמשתמש ארבעה מספרים שלמים, a, b, c ו- d . התוכנית תדפיס שני מספרים קטנים ביותר מתוך הארבעה, בסדר כלשהו.

דוגמאות:

אם הקלט הוא המספרים $a = 2, b = 1, c = 4, d = 3$ התכנית יכולה להדפיס

2

1

או

1

2

אם הקלט הוא המספרים $a = 1, b = 1, c = 4, d = 1$ התכנית תדפיס

1

1

אתם מתבקשים לתת פתרון מבוסס פעולות של שיחלוף-מותנה. שימו לב: בקוד שתכתבו, אין להשתמש במשפטי if או if-else שלא בהקשר של פעולת שיחלוף-מותנה. עובדה מתמטית: ניתן לפתור משימה זו עם ארבע פעולות שיחלוף-מותנה, אך לא עם פחות מכך.

ההערות הנוספות (איך לכתוב הערות וכו') לא רלוונטיות אלינו.

שימו לב: במשימה זו, בקובץ Task6.java, יש לרשום את כל ההערות בצורה // לדוגמה:

```
// This is a comment
```

אין לרשום הערות מהצורה

```
/* Do not write this way */
```

פרט לכך, מותר לרשום שורות קוד אך ורק בשורות שבין ההערות שבקובץ:

```
// ----- "A": write your code BELOW this line only -----  
// your code here (add lines)
```

מותר לרשום שורות קוד רק כאן.
אסור לשנות את הקובץ בכל איזור אחר.

```
// ----- "B" write your code ABOVE this line only -----
```

משימה 7 – תוכנית בדיקה (tester) (15 נקודות)

במשימה זו תכתבו תוכנית אשר תבדוק את נכונות התוכנית שכתבתם במשימה הקודמת. התוכנית מהמשימה הקודמת תחשב נכונה אם עבור כל קלט למשתנים a, b, c ו- d , הערכים המודפסים הינם שני מספרים קטנים ביותר מתוך הארבעה. אם למשל, עבור הקלטים $a=1, b=0, c=0, d=1$ התוכנית מהמשימה הקודמת תדפיס את השורות:

0

1

אזי התוכנית אינה נכונה. במקרה זה נאמר ש $a=1, b=0, c=0, d=1$ היא "דוגמה נגדית" לנכונות התוכנית.

תוכלו להעזר בעובדה המתמטית הבאה:

התוכנית שכתבתם במשימה 6 נכונה, אם היא עובדת נכון עבור כל הקלטים שבהם המשתנים a, b, c ו- d מקבלים ערכי אפס או אחד. כלומר, במקרה שלנו, מספיק לבדוק 2^4 אפשרויות כדי לוודא שהתוכנית נכונה.

מי שמתעניין בהוכחה לטענה זו מוזמן לעיין ב <https://mitpress.mit.edu/sites/default/files/Chapter%2027.pdf>.

כמובן שלא נרצה לבדוק את נכונות התוכנית על-ידי הקלדת כל הקלטים האפשריים ובחינה של כל הפלטים המתקבלים.

התוכנית שתכתבו במשימה זו תייצר קלטים (במקום לקבלם מהמשתמש) ותדפיס לפי אחת משתי האפשרויות הבאות:

א. אם התוכנית מהמשימה הקודמת אכן נכונה, עליכם להדפיס במשימה זו את המילה `verified`.

ב. אם התוכנית מהמשימה הקודמת אינה נכונה, עליכם להדפיס ערכים של a, b, c ו- d שהם דוגמה נגדית לחוסר

הנכונות. יש להדפיס רק דוגמה נגדית אחת, אפילו אם יש דוגמאות רבות. זכרו - כל ערך צריך להיות מודפס

בשורה נפרדת.

שימו לב: במשימה זו, בקובץ Task7.java, מותר לרשום קוד כרצונכם, פרט למקום המסומן בקובץ שלשם אתם חייבים להעתיק שורות ממשימה מספר 6. בהנחה שהקוד שכתבתם בסעיף הקודם הוא נכון, אזי הפלט של התכנית במשימה זו ידפיס את המילה verified ואחרת תדפיס דוגמה נגדית. אנו נבדוק משימה זו גם על הפתרון שלכם למשימה 6 וגם על אוסף של פתרונות אחרים, נכונים ולא נכונים, עבור משימה 6.

מבנה הקובץ Task7.java מתואר כאן. הקפידו לכתוב את הפתרון שלכם לפי המבנה הנדרש.

```
// ----- write any code BELOW this line only -----  
// your code here (add lines)
```

מותר לכתוב כאן קוד

```
// ----- write any code ABOVE this line only -----
```

```
// ----- copy here the code from Task 9 that is between  
// ----- the comments "A" and "B"  
// code from Task 6 here
```

העתיקו לכאן בדיוק את קטע הקוד ממשימה 6

```
// ----- end of copied code from Task 6
```

```
// ----- write any code BELOW this line only -----  
// your code here (add lines)
```

מותר לכתוב כאן קוד

```
// ----- write any code ABOVE this line only -----
```

במשימה זו יש לרשום את כל ההערות בצורה
לדוגמה:

```
// This is a comment
```

אין לרשום הערות מהצורה

```
/* Do not write this way */
```

אין לשנות את הקובץ Task7.java פרט להוספת קוד במקומות המותרים.

בהצלחה!