

מבוא למדעי המחשב - סמסטר א' תשע"ט

עבודת בית מספר 1

צוות התרגיל: מיכאל פרנק, יערה שובל, מיכל שמש

תאריך פרסום: 28.10.17

תאריך הגשה: 8.11.17, 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 ביחידים. 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.
3. לעבודה מצורפים קובצי Java עם שמות כגון `Task<n>.java`, כאשר `<n>` מציין 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 10 משימות וסך הנקודות המקסימלי הוא 110. כלומר, יש אפשרות לקבל 10 נקודות בונים. שימו לב שמספר הנקודות לכל משימה אחיד (11 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 (עבור עצמכם) במערכת ההגשות.

שלשות פיתגוריות

הגדרה: (שלשה פיתגורית)

שלשה מספרים שלמים a, b, c נקראים שלשה פיתגורית אם מתקיים כי $0 < a \leq b \leq c$ וגם $a^2 + b^2 = c^2$.

משימה 1א - משימת חימום (11 נקודות)

פתחו את הקובץ Task1a.java וכתבו בו תכנית אשר קולטת מהמשתמש שלושה מספרים שלמים ומדפיסה למסך "yes" אם שלושת המספרים הם שלשה פיתגורית ו-"no" אחרת. להזכירכם, **אין** להדפיס דברים מיותרים למסך (כגון: "please enter three integers").

דוגמאות:

אם הקלט הינו $a = 3, b = 4, c = 6$ אזי הפלט יהיה:

no

אם הקלט הינו $a = 3, b = 4, c = 5$ אזי הפלט יהיה:

yes

אם הקלט הינו $a = 3, b = -4, c = 6$ אזי הפלט יהיה:

no

אם הקלט הינו $a = 4, b = 3, c = 5$ אזי הפלט יהיה:

no

כיוון ש- $4 > 3$

אם הקלט הינו $a = 0, b = 2, c = 4$ אזי הפלט יהיה:

no

כיוון ש- $a \leq 0$

משימה 1ב (11 נקודות)

פתחו את הקובץ Task1b.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם אי-שלילי ומדפיסה למסך את כל השלשות הפיתגוריות שאיבריהן קטנים ממספר זה. כלומר, אם המספר שנקלט הינו n יש להדפיס את כל השלשות הפיתגוריות $0 \leq a \leq b \leq c \leq n$.

דוגמא:

אם הקלט הוא 24, אזי הפלט יהיה:

```
3 4 5
5 12 13
6 8 10
8 15 17
9 12 15
12 16 20
```

יש להדפיס כל שלשה בשורה נפרדת. כל שורה תכיל את שלושת המספרים המרכיבים את השלשה עם רווחים ביניהם. **אין** להדפיס רווח לאחר המספר השלישי, כלומר, יש להדפיס את המספר הראשון, אחריו רווח בודד, אחריו את המספר השני, אחריו רווח בודד, ולסיום את המספר השלישי. ניתן להניח כי הקלט הינו מספר שלם אי-שלילי. להזכירכם, **אין** להדפיס דברים מיותרים למסך (כגון: "please enter three integers").

עצרת ושארית חלוקה

משימה 2א: עצרת (11 נקודות)

הגדרה: (עצרת)

עבור מספר שלם אי-שלילי $n \geq 0$ העצרת של n היא מכפלת כל המספרים הקטנים או שווים ל- n . העצרת של 0 מוגדרת להיות 1. נסמן את העצרת $n! = 1 * 2 * 3 * \dots * n$.

פתחו את הקובץ Task2a.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם n ומדפיסה למסך את הערך של $n!$, כלומר את העצרת של n .

דוגמאות:

אם הקלט הינו $n = 0$ אז התכנית תדפיס:

1

אם הקלט הינו $n = 5$ אז התכנית תדפיס:

120

אם הקלט הינו $n = 12$ אז התכנית תדפיס:

479001600

אם הקלט הינו $n = 18$ אז התכנית תדפיס:

-898433024

נסו להבין מדוע?

יש להניח כי המספר n הינו שלם אי-שלילי. על התכנית לחשב נכונה את העצרת של המספרים בין 0 ל-12 כולל.

שימו לב: עליכם לחשב את $n!$ רק לאחר קבלת הקלט מהמשתמש. תשובות אשר יחשבו מראש את הערכים האפשריים ל- n לא יתקבלו.

משימה 2ב: שארית חלוקה (11 נקודות)

תזכורת: (שארית חלוקה)

שארית החלוקה של n ב- k הינה מספר r בין 0 ל- $k - 1$ כך שקיים מספר m עבורו $n = k * m + r$.

פתחו את הקובץ Task2b.java וכתבו בו תכנית אשר קולטת מהמשתמש שני מספרים שלמים n, k ומדפיסה למסך את הערך של $n \% k$, כלומר את שארית החלוקה של העצרת של n ב- k . המספר הראשון שיש לקלוט הינו n והמספר השני שיש לקלוט הינו k , בסדר הזה.

דוגמאות:

אם בקלט שני הערכים הם $n = 12, k = 67$ אזי הפלט יהיה:

41

כיוון ש- $12! = 479001600$ ומתקיים ש- $12! = 67 * 7149277 + 41$

אם בקלט שני הערכים הם $n = 18, k = 71$ אזי הפלט יהיה:

56

כיוון ש- $18! = 6402373705728000$ ומתקיים ש- $18! = 71 * 901742277545464 + 56$

יש להניח כי המספרים n, k הינם שלמים אי-שליליים וכי $k > 1$.
על התכנית לחשב נכונה את $n! \% k$ לכל ערך של n ו- k .

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

משימה 3 – הדפסת הגורמים הראשוניים של מספר נתון (11 נקודות)

פתחו את הקובץ Task3.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. על כל שורה בפלט להכיל את הגורם הראשוני ואת מספר הפעמים שאותו גורם מופיע במכפלה, במידה ואותו גורם מופיע יותר מפעם אחת במכפלה. להזכירכם, אין להדפיס דברים מיותרים למסך (כגון: "please enter an integer").

דוגמאות:

אם הקלט הוא 100, הפלט יהיה:

2 2

5 2

כיוון שהגורם הראשוני 2 מופיע פעמיים והגורם הראשוני 5 מופיע פעמיים בפירוק הראשוני של 100.

אם הקלט הוא 6, הפלט יהיה:

2

3

אם הקלט הוא 23, הפלט יהיה:

23

אם הקלט הוא 8112, הפלט יהיה:

2 4

3

13 2

סיכום שברים

משימה 4א – חישוב מחלק משותף מקסימלי ל-3 מספרים (11 נקודות)

פתחו את הקובץ Task4a.java וכתבו בו תכנית אשר קולטת מהמשתמש שלושה מספרים שלמים וחיוביים a, b, c ומדפיסה את המחלק המקסימלי המשותף הגדול ביותר של a, b, c .

עובדה מתמטית: המחלק המשותף המקסימלי של שלושה מספרים a, b, c מסומן $\gcd(a, b, c)$ ומוגדר כ-
$$\gcd(a, b, c) = \gcd(\gcd(a, b), c)$$

דוגמאות:

אם הקלט הוא המספרים 18, 24, 36, הפלט יהיה
6

כיוון ש-6 הוא המחלק המקסימלי המשותף הגדול ביותר עבור שלושת מספרים אלו.

אם הקלט הוא המספרים 12, 18, 25, הפלט יהיה
1

כיוון שאין מספר שונה מ-1 המחלק את שלושת מספרי הקלט.

משימה 4ב – חישוב כפולה משותפת מינימלית ל-3 מספרים (11 נקודות)

הגדרה: הכפולה המשותפת המינימלית של שני מספרים a, b מסומנת באמצעות $\text{lcm}(a, b)$ והיא המספר השלם החיובי הקטן ביותר ששני המספרים a, b מחלקים אותו

עובדה מתמטית:

$$\text{lcm}(a, b) = \frac{a * b}{\gcd(a, b)}$$

הגדרה: הכפולה המשותפת המינימלית של שלושה מספרים a, b, c מסומנת באמצעות $\text{lcm}(a, b, c)$ ומוגדרת כ-

$$\text{lcm}(a, b, c) = \text{lcm}(\text{lcm}(a, b), c)$$

שימו לב:
$$\text{lcm}(a, b, c) \neq \frac{a * b * c}{\gcd(a, b, c)}$$

פתחו את הקובץ Task4b.java וכתבו בו תכנית אשר קולטת מהמשתמש שלושה מספרים שלמים וחיוביים a, b, c ומדפיסה את הכפולה המשותפת המינימלית של a, b, c .

דוגמאות:

אם הקלט הינו $a = 16, b = 18, c = 6$ התכנית תדפיס

144

כיוון ש- $144 = 6 * 24 = 18 * 8 = 16 * 9$

ו-144 הינו המספר השלם המינימלי שהוא כפולה שלמה של 6, 16 ו-18.

אם הקלט הינו $a = 2, b = 3, c = 11$ התכנית תדפיס

66

כיוון ש- $66 = 2 * 33 = 3 * 22 = 11 * 6$

ו-66 הינו המספר השלם המינימלי שהוא כפולה שלמה של 2, 3 ו-11.

משימה 4g – סיכום שברים (11 נקודות)

פתחו את הקובץ Task4c.java וכתבו בו תכנית אשר קולטת מהמשתמש שישה מספרים שלמים וחיוביים a, b, c, d, e, f .
על התכנית לבדוק האם:

$$\frac{a}{b} + \frac{c}{d} + \frac{e}{f} = 1$$

ובמידה שכן, להדפיס למסך "yes", במידה שלא להדפיס למסך "no". ניתן להניח כי $a, b, c, d, e, f \geq 0$ וכי $b, d, f > 0$.

שימו לב: ניתן להגדיר במשימה זו אך ורק משתנים מטיפוס int.

דוגמאות:

אם $a = 1, b = 2, c = 1, d = 4, e = 1, f = 4$ יש להדפיס למסך

yes

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

אם $a = 1, b = 2, c = 1, d = 3, e = 1, f = 6$ יש להדפיס למסך

yes

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

אם $a = 100000, b = 200000, c = 100000, d = 300000, e = 10000, f = 600000$ יש להדפיס

למסך

yes

כיוון ש- $\frac{100000}{200000} + \frac{100000}{300000} + \frac{100000}{600000} = 1$

אם $a = 1, b = 2, c = 1, d = 4, e = 1, f = 3$ יש להדפיס למסך

no

כיוון ש- $\frac{1}{2} + \frac{1}{4} + \frac{1}{3} \neq 1$

אם $a = 100000$, $b = 200000$, $c = 100000$, $d = 400000$, $e = 100000$, $f = 300000$ יש להדפיס
למסך:

no

כיוון ש- $1 \neq \frac{100000}{200000} + \frac{100000}{400000} + \frac{100000}{300000}$

בחירה על ידי חילוף-מותנה

בשתי המשימות הבאות נעסוק באלגוריתמי בחירה המוגדרים במונחים של פעולה הנקראת "חילוף-מותנה". חילוף-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 החילוף-המותנה שבתכנית.

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

משימה 5 – מציאת מספר קטן ביותר וגדול ביותר (11 נקודות)

פתחו את הקובץ `Task5.java` וכתבו בו תכנית אשר מקבלת מהמשתמש חמישה מספרים שלמים, `a, b, c, d` ו-`e`. התוכנית תדפיס את המספר הקטן ביותר, והגדול ביותר מבין החמישה, בסדר זה.

דוגמאות:

אם הקלט הוא המספרים $a = 2, b = 1, c = 3, d = 2, e = 1$ התכנית תדפיס

1
3

אם הקלט הוא המספרים $a = 2, b = 1, c = 1, d = 17, e = -1$ התכנית תדפיס

-1
17

אתם מתבקשים לתת פתרון מבוסס פעולות של חילוף-מותנה. כלומר, על הפתרון של משימה 5 להכיל פעולות קלט, פעולות פלט ופעולות חילוף מותנה בלבד. בפרט, בקוד שתכתבו, אין להשתמש במשפטי `if` או `if-else` שלא בהקשר של פעולת חילוף-מותנה.

עובדה מתמטית: ניתן לפתור משימה זו עם שש פעולות חילוף-מותנה, אך לא עם פחות מכך.

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

שימו לב: במשימה זו, בקובץ Task5.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 6 – תוכנית בדיקה (tester) (11 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, d, e הערכים המודפסים הינם הערך המינימלי והמקסימלי מבין החמישה. אם למשל, עבור הקלטים $a=1, b=0, c=0, d=1, e=0$ התוכנית מהמשימה הקודמת תדפיס את השורות:

0

0

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

תוכלו להעזר בעובדה המתמטית הבאה:

התוכנית שכתבתם במשימה 5 נכונה, אם היא עובדת נכון עבור כל הקלטים שבהם המשתנים a, b, c, d, e מקבלים ערכי אפס או אחד. כלומר, במקרה שלנו, מספיק לבדוק 2^5 אפשרויות כדי לוודא שהתוכנית נכונה. מי שמתעניין בהוכחה לטענה זו מוזמן לעיין ב:

1. https://en.wikipedia.org/wiki/Sorting_network#Zero-one_principle
2. [Knuth, D. E. \(1997\). *The Art of Computer Programming*, Volume 3: *Sorting and Searching* \(Second ed.\). Addison–Wesley. pp. 219–247. ISBN 0-201-89685-0.](#) Section 5.3.4: Networks for Sorting.
3. [Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. \(1990\). *Introduction to Algorithms* \(1st ed.\). MIT Press and McGraw-Hill. ISBN 0-262-03141-8.](#)

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, d, e 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 Task6.java, מותר לרשום קוד כרצונכם, פרט למקום המסומן בקובץ שלשם אתם חייבים להעתיק שורות ממשימה מספר 5. 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 5 וגם על אוסף של פתרונות אחרים, נכונים ולא נכונים, עבור משימה 5.

מבנה הקובץ Task6.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 5 that is between  
// ----- the comments "A" and "B"  
// code from Task 5 here
```

העתיקו לכאן בדיוק את קטע הקוד ממשימה 5

```
// ----- end of copied code from Task 5
```

```
// ----- 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 Task6.java פרט להוספת קוד במקומות המותרים.

בהצלחה!