

מבוא למדעי המחשב – עבודה 1

הוראות מקדימות :

1. העבודה כתובה בלשון זכר (מטעמי נוחיות), אך פונה לשני המינים.
2. ניתן ומומלץ לבצע את העבודה בזוגות בהתאם להוראות המופיעות בסילבוס הקורס.
3. לעבודה מצורפים קבצי java, עם שמות כגון "Ex<num>.java", כאשר "<num>" מציין את מספר המשימה המתאימה לקובץ. צרו תיקייה חדשה והעתיקו את קבצי ה-java אליה. עליכם לערוך קבצים אלו בהתאם למפורט בתרגיל, ולהגיש את קבצי ה-java הערוכים בלבד, מכווצים כקובץ ZIP יחיד. אין לשנות את שמות הקבצים, להגיש קבצים נוספים, ליצור תיקיות, להגיש מספר קבצים עבור אותה המשימה, או להגיש קובץ יחיד למספר משימות. שם הקובץ יכול להיות על פי שיקולכם ובאנגלית בלבד. **קבצים שיוגשו בפורמט שונה מ- zip לא ייבדקו.**
4. את הקובץ יש להגיש ב Submission System.
5. העבודה תיבדק באופן אוטומטי לפי הפלט אשר התוכניות שלכם תדפסנה למסך. לכן, יש להקפיד על ההוראות ולבצע אותן **במדויק**. כל הדפסה אשר אינה עונה **באופן מדויק** לדרישות המופיעות בעבודה (כולל שורות, רווחים, סימני פיסוק, או כל תו אחר מיותרים, חסרים, או מופיעים בסדר שונה מהנדרש), תגרור פגיעה משמעותית בציון.
6. סגנון כתיבת הקוד ייבדק באופן ידני. יש להקפיד על כתיבת קוד ברור, מתן שמות משמעותיים למשתנים, הזחות (אינדנטציה), **והוספת הערות בקוד** המסבירות את תפקידם של מקטעי קוד שונים. אין צורך למלא את הקוד בהערות סתמיות אך חשוב לכתוב הערות בנקודות קריטיות המסבירות קטעים חשובים בקוד. כתיבת קוד אשר אינו עומד בסטנדרטים אלו תגרור הפחתה בציון העבודה. בנוסף, הערות יש לרשום אך ורק באנגלית (כתיבת הערות בכל שפה אחרת שקולה לאי כתיבת הערות).
7. בכדי לוודא שהגשתם את התרגיל באופן תקין – הורידו את קובץ הZIP שהגשתם ב Submission System למחשב שלכם, חלצו אותו ונסו להדר ולהריץ את התוכנית. הליך זה הוא חשוב, עקב מקרים שקרו בעבר בהם הקובץ שהועלה למערכת ההגשות היה פגום.
8. במידה ואינכם בטוחים מהו הפירוש המדויק להוראה מסוימת, או בכל שאלה אחרת הקשורה **בתוכן** העבודה, אנא היעזרו בפורום או בשעות הקבלה של האחראים על העבודה (פרוט שעות הקבלה מופיע באתר הקורס). בכל בעיה **אישית** הקשורה בעבודה (מילואים, אשפוז וכו'), אנא צרו את

- הפניה המתאימה במערכת הגשת העבודות כפי שמוסבר בסילבוס שבאתר הקורס (כאן).
9. בכל פעם שאתם מתבקשים להדפיס למסך, עליכם להשתמש בפונקציה `System.out.println`, אשר מדפיסה למסך ויורדת שורה (לכן כל פעולת הדפסה תופיע בשורה נפרדת).
10. בכדי לקבל נתונים מהמשתמש יש להשתמש ב-Scanner, כפי שנלמד בכיתה.
11. שימו לב כי בעבודה ניתנו 5 נקודות "תמריץ" עבור מעקב אחר הוראות העבודה. במידה וההוראות לא מולאו (שכחתם להוסיף שותף, חוסר בדיקה של פורמט הקבצים, שמות לא נכונים וכדומה) הדבר יגרור הורדה של נקודות אלו.
12. סך כל הנקודות בעבודה מסתכם ל-100 נקודות כאשר:
- 12.1. 5 נקודות עבור מעקב אחר הוראות (סעיף 11)
- 12.2. 15 נקודות עבור הבדיקה הידנית - כתיבת הערות (7 נק'), הזחה (4 נק') ושמות משתנים (4 נק').
- 12.3. 80 נקודות על כתיבת הקוד בהתאם לדרישות התרגיל
13. לא ניתן להשתמש במערכים, מחרוזות, פונקציות, או כל צורת קוד אחרת אשר לא נלמדה בכיתה. שימוש באחת מצורות הקוד הנ"ל עלול לגרור הפחתה בציון העבודה.

הימנעו מהעיתקות! ההגשה היא בזוגות. במידה ומוגשות עבודות בנפרד עם קוד זהה או אפילו דומה – זוהי העתקה אשר תדווח לוועדת משמעת לאלתר.

אנא קראו את סילבוס הקורס שנית !

משימה 0 – הכרת מערכות הקורס ועבודה בסיסית בJava

סעיף א'

העבודה בזוגות על מחשב במעבדה או מחשבו האישי של אחד הסטודנטים. ההוראות מתוארות כפי שיבוצעו ע"י סטודנטים שיבחרו לבצען במעבדה.

מטרות: תרגול פעולות מחשב בסיסיות, עבודה עם חלון ה-command, הגשת עבודה משותפת.

- צרו ספרייה בשם Test (בחשבונכם האישי בכונן S).
- הורידו מאתר הקורס את הקובץ `Main.java` לתוך הספרייה Test.

- פתחו את הקובץ Main.java בעזרת התוכנה ++Notepad.
- החליפו את המילים "Hello World" במילים "Pikachu rules".
- כנסו בחלון ה-command לכוון האישי שלכם (S) ולספריה Test. (להסברים על ה-command אנא הסתכלו תחילה במצגת התרגול הראשונה הנמצאת באתר הקורס).
- הדרו (compile) את הקובץ Main.java המופיע בספריה. שימו לב, בקובץ זה קיימת שגיאה במתכוון. עליכם לזהותה ולתקנה. אם ביצעתם את התיקון כראוי, הקוד יתקמפל בהצלחה. ודאו כי הקומפילציה הצליחה וכי נוצר הקובץ Main.class.
- הריצו את הקובץ Main. אם ביצעתם את השינוי והתיקון כראוי לאחר הריצה יופיעו המילים "Pikachu rules".
- צרו קובץ zip המכיל רק את הקובץ Main.java ללא קובץ class שנוצר כתוצאה מההידור (ולאחר שביצעתם את התיקון הנדרש).
- בצעו כניסה למערכת הגשת העבודות:
<http://www.cs.bgu.ac.il> -> Education -> Useful links -> Submission System
או דרך הקישור באתר הקורס משמאל (בסוף המשימה מופיע הסבר מה לעשות במקרה שאתם לא מצליחים להתחבר למערכת).
- בחרו בחלון הקורסים את הקורס "מבוא למדעי המחשב".
- היכנסו ל-View grades, כעת מופיעות המילים Not registered, עבור התרגילים במשימה Pikachu. היכנסו ל-Register group, כעת יצרתם קבוצת הגשה למשימה זו אשר תכיל רק אתכם. עתה, חפשו את שם המשתמש של השותף שלכם, אם אינו מופיע אזי קיימות מספר אפשרויות:
 - הוא כבר רשום למשימה זו. עליו להיכנס למערכת ולהוריד עצמו מקבוצת ההגשה שלו למשימה זו. כעת שמו יעלם מראש הרשימה ויחזיר אותו לסטטוס של סטודנט ללא קבוצה. במצב זה תוכלו אתם לראותו ברשימה שלכם, לאחר שתכנסו לתרגיל פעם נוספת.
 - הוא אינו רשום כלל למערכת הגשת העבודות, יש צורך לפנות לאחד מאנשי סגל הקורס.

סמנו את שמו של שותפכם, הקלידו את מספר תעודת הזהות שלו ולחצו על Add partner, כעבור מספר שניות תופיעו שניכם בראש הרשימה. אם הפעולה לא הצליחה, יש לוודא כי אחד השותפים מחק את עצמו מקבוצתו. זאת, כי לא ניתן להוסיף סטודנט לקבוצה אם אותו סטודנט כבר רשום לקבוצה אחרת (אפילו אם אותה קבוצה מכילה סטודנט יחיד).

- בקשו משותפכם לבצע כניסה מחדש למערכת. ע"י לחיצה על View grades יוכל לראות כי גם הוא רשום לקבוצת ההגשה למשימה זו, כאשר מספר קבוצת ההגשה שלכם זהה. מרגע הפיכתכם לזוג כל פעולה שתבצע ע"י אחד מכם תראה גם במערכת של בן הזוג לכן אין צורך לבצע פעולות פעמיים. כמו כן, לא משנה באיזה מחשב מתבצעות הפעולות.
- שילחו את קובץ ה-zip שיצרתם, תוכלו לראות כי התווסף קישור של העבודה כולל תאריך הגשה. היכנסו ל-View grades ותראו כי המילים Submitted (או Late submit, במקרה של הגשה מאוחרת) מופיעות. שימו לב כי ישנה אפשרות לסירוב הגשה, במקרה של הגשה בעייתית.
- הגישו את קובץ ה-zip עוד פעם, כדי לראות שניתן להגיש את העבודה מס' פעמים (ההגשה האחרונה היא הקובעת).

* אם אינכם מצליחים להיכנס למערכת, אנא שלחו email לצוות המתרגלים האחראיים על התרגיל. את כתובות email של כל חברי צוות הקורס ניתן למצוא בעמוד של כל חבר צוות באתר המחלקה. ניתן להגיע אליו מקישור באתר המחלקה או מקישור שניתן למצוא באתר הקורס תחת Course info.

משימה 1 - חימום קל והבנת הנקרא (5 נק')

- קראו את ההוראות פעם נוספת וודאו כי הבנתם אותן כראוי!
- קראו שוב את סעיפים 5 ו-6 בהוראות.

- פתחו את הקובץ Ex1.java .
- כתבו תוכנית אשר מדפיסה את כל המספרים המתחלקים ב-4 מ-4 ועד 648 בסדר עולה (כולל 4 ו-648). שימו לב כי לפי ההוראות, כל מספר צריך להופיע בשורה חדשה.
- פתחו את הקובץ Ex1Test.java והריצו את התוכנית. תוכנית זאת בודקת את התוכנית שכתבתם. במידה והצלחתם יוצג הכיתוב "GOOD WORK", ותוכלו להמשיך למשימה הבאה, במידה ולא – חזרו על ההוראות ונסו שוב.

משימה 2 – מספרים ראשוניים (10 נק')

פתחו את הקובץ Ex2.java. כתבו תוכנית אשר מקבלת מהמשתמש שני מספרים שלמים גדולים ממש-0, ומדפיסה למסך את הממוצע החשבוני של כל המספרים הראשוניים הנמצאים בטווח המספרים (כולל הקצוות). אם לא קיימים מספרים ראשוניים בטווח הנ"ל, על התוכנית להדפיס 0.

ניתן לקרוא על מספרים ראשוניים [כאן](#) ועל ממוצע חשבוני [כאן](#).

ניתן להניח כי המשתמש יכניס מספרים שלמים וגדולים ממש-0. להזכירכם, אין להדפיס דברים מיותרים למסך (כגון: "please enter an integer").

דוגמאות:

אם המשתמש הכניס את המספרים 2 ו-11, הפלט יהיה 5.6 (מכיוון ש-2, 3, 5, 7 ו-11 הם הראשוניים שנמצאים בתחום [2,11], והממוצע החשבוני שלהם הוא $28 / 5 = 5.6$)

משימה 3 – ממוצע שלושת הגדולים (20 נק')

פתחו את הקובץ Ex3.java. כתבו תוכנית אשר מקבלת מהמשתמש סדרה של מספרים שלמים, שונים זה מזה וגדולים ממש-0, ומדפיסה למסך את ממוצע שלושת המספרים הגדולים מבין כל המספרים. שימו לב כי לא ידוע כמה מספרים המשתמש יכניס, ועל

התוכנית לקבל קלטים עד אשר המשתמש הכניס את המספר 0. ניתן להניח כי המשתמש יכניס מספרים שלמים, גדולים ממש מ-0 (מלבד המספר האחרון השווה ל-0) וכי יוכנסו לפחות שלושה מספרים שונים מלבד 0.

תזכורת: אין להשתמש במערכים ! שימוש במערכים יגרור ציון אפס על תרגיל זה.

דוגמה:

אם המשתמש הכניס את המספרים הבאים:

11

3

42

34

2

21

9

0

על הפלט להיות 32.333332 ((21 + 34 + 42) / 3).

משימה 4 – התאמת פלט (15 נק')

פתחו את הקובץ Ex4.java. כתבו תוכנית אשר מקבלת מהמשתמש מספר שלם x , ומדפיסה

את כל השלשות הפיתגוריות (a, b, c) המקיימות:

$$0 < a < b < c < x$$

* ניתן לקרוא בנוסף על שלשות פיתגוריות [כאן](#).

יש להדפיס את השלשות מסודרות בסדר עולה לפי המספר a , כאשר כל שלשה תודפס בדיוק פעם אחת. במידה וישנן שתי שלשות שונות בעלות a זהה, על השלשה בעלת ה- b הקטן יותר להופיע קודם.

פורמט כל שורה (שלשה) בפלט **חייב** להיות זהה (!!!) לפורמט הבא:

$$(a,b,c) : a*a + b*b = c*c$$

הקווים התחתונים האדומים בפורמט נועדו לסמן את הרווחים. אין להדפיס את הקווים למסך, אלא להדפיס במקומות אלו רווחים. הדפסה לא נכונה של רווחים תגרור פגיעה משמעותית בציון!

ניתן להניח כי המשתמש יכניס מספר שלם וגדול ממש 0.

דוגמה: אם המשתמש הכניס את המספר 30, על הפלט להיות:

```
(3,4,5) : 3*3 + 4*4 = 5*5
(5,12,13) : 5*5 + 12*12 = 13*13
(6,8,10) : 6*6 + 8*8 = 10*10
(7,24,25) : 7*7 + 24*24 = 25*25
(8,15,17) : 8*8 + 15*15 = 17*17
(9,12,15) : 9*9 + 12*12 = 15*15
(10,24,26) : 10*10 + 24*24 = 26*26
(12,16,20) : 12*12 + 16*16 = 20*20
(15,20,25) : 15*15 + 20*20 = 25*25
(20,21,29) : 20*20 + 21*21 = 29*29
```

משימה 5 – פירוק ספרות (15 נק')

פתחו את הקובץ Ex5.java. כתבו תוכנית אשר מקבלת מהמשתמש מספר שלם גדול או שווה ל-1, ומדפיסה את סכום כל המספרים השלמים הגדולים מאפס וקטנים או שווים למספר שנקלט, שסכום ספרותיהם מתחלק ב-7 ללא שארית. ניתן להניח כי המשתמש יכניס מספר שלם, גדול או שווה ל-1. רמז: חישבו כיצד ניתן להיעזר בפעולת המודולו (חישוב שארית חלוקה, %) בכדי לחשב את ערכה של ספרה מסוימת במספר (בייצוג עשרוני).

דוגמה:

אם המשתמש הכניס את המספר 30, על הפלט להיות 48 (שכן 7, 16, 25 הם המספרים היחידים שקטנים או שווים ל-30 שסכום ספרותיהם מתחלק ב-7 ללא שארית, וכמו כן

מתקיים $7+16+25 = 48$

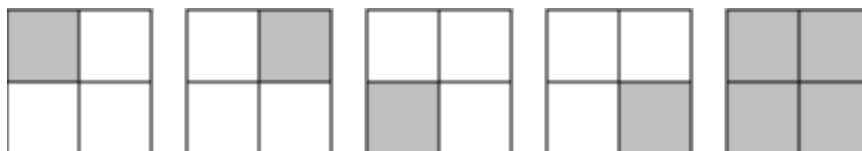
משימה 6 – מניית ריבועים (15 נק')

במשימה זו עליכם לכתוב תוכנית אשר מונה את מספר הריבועים הקיימים בלוח בעל $n \times n$ ריבועים. פתחו את הקובץ Ex6.java. על התוכנית לקבל מהשתמש מספר שלם n גדול או שווה ל-1 המייצג את כמות הריבועים שיש בכל שורה ובכל עמודה בלוח בגודל $n \times n$. על התוכנית להדפיס את כמות הריבועים הקיימים בלוח שכזה.

דוגמה:

בלוח בגודל 2×2 , על הפלט להיות: 5

(ישנם 5 ריבועים שונים - 4 בגודל 1×1 ואחד בגודל 2×2 , כפי שמודגם באיור המצורף)



שימו לב - עליכם למנות רק ריבועים (לא מרובעים).

בהצלחה!!!