

תוכנה 1 – אביב 2022/23

תרגיל מספר 2

הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.
את התרגיל הבא צריך להגיש באופן הבא:

- הגשה במערכת ה-Git תבצע על פי ההנחיות שראיתם בתרגול 1.
צרו את ה repository שלכם מתוך הקישור הבא:

<https://classroom.github.com/a/lc4CngJ5>

יש לוודא שבתקיית הגיט שלכם נמצאים הקבצים הבאים:

- a. קובץ פרטים אישיים בשם details.txt המכיל את שם המשתמש שלכם ב Moodle ואת מספר תעודת הזהות שלכם.
- b. קבצי ה- java של התוכניות אותם התבקשתם לממש. בתרגיל הנוכחי ישנם 5 קבצים java אשר נמצאים בתקיית src.
- הגשה במערכת ה Moodle (<http://moodle.tau.ac.il/>): עליכם להגיש את קובץ הטקסט assignment.txt ובו קישור ל repository git האישי שלכם. הקובץ צריך להכיל שורה אחת בדיוק, ללא מלל נוסף. לדוגמא, עבור תרגיל 2 הקובץ יכול את השורה הבאה, כשבמקום githubUser יופיע המשתמש שלכם ב github:

<https://github.com/software1course2223a/hw2-githubUser.git>

שימו לב: חשוב מאד להקפיד על פורמט ההגשה. כלומר, יש לקרוא לקבצים בדיוק לפי ההנחיות שקיבלתם. כמו כן אין לצרף תיקיות או קבצים נוספים. **אי עמידה בהנחיות ההגשה תגרור הורדה משמעותית בניקוד!**

נדגיש שוב: בתקיית הגיט שלכם יופיעו קובץ details.txt, ותיקיית src שבתוכה קובץ Assignment1.java. בנוסף, יופיעו קבצי מערכת כגון .gitignore ו-README.

לפני ההגשה, עליכם להריץ כל תכנית על מספר קלטים שונים (ובפרט לפי הדוגמאות המפורטות בתרגיל) כדי לוודא שהיא אכן פועלת כראוי.

שימו לב – חשוב לוודא כי הקוד רץ תקין על שרת ה-nova בהתאם להוראות אשר ניתנו בהוראות תרגיל בית 1.

ניתן להניח כי הקלט תקין (מספר ארגומנטים בקלט\המרה לטיפוסים וכו'), אלא אם צוין אחרת. במקרה שתצטרכו להתמודד עם קלט לא תקין, תופיע הנחיה בתרגיל על אופן הטיפול הרצוי. בנוסף, כל הקלטים בתרגיל זה מתקבלים בשורת הפקודה, כלומר, ארגומנטים לפונקציה main.

שאלה 1

[20 נק'] חימום:

כפי שראיתם בשבוע הראשון של הקורס, בג'אווה כל char מיוצג ע"י ערך מספרי. כדי לדעת מהו הערך של כל תו בג'אווה צריך להסתכל [בטבלת אסקי ascii table](#), שזהו סטנדרט בינלאומי למיפוי אותיות למספרים. כך לפי טבלת ascii הערך של התו 'A' הוא 65, של 'b' הוא 98 והערך של '!' הוא 33.

ממשו את התוכנית Assignment02Q01 אשר מקבלת כקלט מספר כלשהו של מחרוזות. עבור כל מחרוזת התוכנית בוחנת את התו הראשון, ובמידה שהערך ה ascii שלו מתחלק ב 5 ללא שארית, התוכנית מדפיסה את התו בשורה נפרדת. ניתן להיעזר באופרטור [%](#). לדוגמא עבור קלט:

Before A E none

התכנית תדפיס את:

A

n

(כיוון שערך האסקי של B הוא 66, של A הוא 65, של E הוא 69 ושל n הוא 110).

שאלה 2

[20 נק'] ממשו את התוכנית Assignment02Q02 שמחשבת קירוב למספר pi.

התכנית מקבלת בשורת הפקודה מחרוזת אשר מייצגת מספר טבעי כלשהו ומחשבת את המספר פאי באמצעות הביטוי המתמטי הבא:

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \dots \right)$$

הארגומנט שהתוכנית מקבלת ייצג את מספר הנסכמים בתוך הסוגריים.

התכנית תדפיס למסך את הערך שהתקבל ואת הערך של המספר PI כפי שניתן על ידי ה-jdk, שלו ניקרא באמצעות הפקודה Math.PI.

לדוגמא עבור הקלט 4, התכנית תחשב את הביטוי: $4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} \right)$, ותדפיס למסך:

2.8952380952380956 3.141592653589793

(שני הערכים מודפסים באותה השורה, מופרדים ברווח יחיד. העזרו בפקודת ההדפסה שמופיעה בשלד התרגיל).

דוגמא נוספת: עבור הקלט 100 התכנית תדפיס למסך:

3.1315929035585537 3.141592653589793

לצורך החישוב השתמשו במשתנים מסוג double (לא float)

שאלה 3

[20 נק'] ממשו את התוכנית Assignment02Q03 אשר מקבלת כקלט מספר טבעי גדול או שווה ל 3, נקרא לו x, ומדפיסה את x האיברים הראשונים של סדרת פיבונאצ'י. לדוגמא, עבור הקלט 5 יודפסו חמשת האיברים הראשונים. בנוסף, יודפס מספר האיברים האי זוגיים מבין x האיברים הללו. דוגמא נוספת עבור הקלט 10:

The first 10 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55

The number of odd numbers is: 7

שימו לב לפורמט ההדפסה:

בשורה הראשונה – יודפס מספר האיברים שיודפסו.
בשורה השניה יודפסו כל האיברים, אחר אחרי השני, מופרדים ברווחים. שימו לב, שלאחר האיבר האחרון לא יודפס רווח.
בשורה האחרונה יודפס מספר האיברים האי זוגיים מבין איברים אלה.
בשלב הקוד נתונות פקודות ההדפסה למלל, ועליכם להשלים רק את הדפסת החישובים.

דוגמא נוספת: עבור המספר 20 פלט התוכנית יהיה (בפורמט הבא):

The first 20 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The number of odd numbers is: 14

שאלה 4

[20 נק'] התבוננו בפונקציית בתוכנית Assignment02Q04 אשר אמורה לייצר מערך עם 20 המספרים הראשונים הראשונים.

הרעיון לבניית המערך: המספר הראשוני הראשון הוא 2, ולכן הוא מוכנס למערך בשלב האתחול. לאחר מכן, כל המספרים הראשונים יהיו אי-זוגיים, ולכן נעבור רק על מספרים אלה ונבדוק את הראשוניות שלהם. כיצד נבדוק אם מספר X כלשהו הוא ראשוני? נבדוק אם X מתחלק ללא שארית במספר ראשוני כלשהו בין 3 ל שורש X (כולל הקצוות) למה עד שורש X? כיון X לא יכול להיות מכפלה של שני מספרים שגדולים ממש משורש X.

התוכנית שנתונה לכם אמורה לממש רעיון זה, אך נפלה בה טעות במימוש ולכן זהו הפלט שמתקבל:

[2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49]

פלט זה מכיל מספר מספרים שאינם ראשוניים (מודגשים בצהוב).

על מנת להבין מה קורה בתוכנית ואיפה יש בעיות, מומלץ להעזר ב debugger.

קראו את [המדריך לשימוש ב- Eclipse של debugger](#), פרקים 1-3.



הריצו את התוכנית במצב דיבאג :

ועקבו אחרי שלבי הריצה, עד שתמצאו את הגורם לבעיה.

תקנו את התוכנית (יתכן שצריך לתקן במספר מקומות) והגישו את התוכנית המתוקנת.

אם הצלחתם, התכנית תדפיס את המערך הנכון:

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71]

שימו לב כי אין אמנם אין מגבלה על אופי התיקון, אך התיקון הנדרש לא דורש שינוי גדול בקוד.

שאלה 5

[20 נק'] ראינו שמערכים יכולים להיות בעלי יותר ממימד אחד. לכן, ניתן להשתמש במערך דו-מימדי על מנת לייצג מטריצה. ממשו את התוכנית Assignment02Q05 אשר עבור מערך דו-מימדי של מספרים מייצג מטריצה ריבועית ומבצעת שתי הדפסות:

1. המטריצה המקורית

2. אותה המטריצה בסיבוב של 90 מעלות ימינה (ראו דוגמאות למטה).

המערך הדו מימדי שמייצג מטריצה נוצר ע"י הקלט לתוכנית. מטריצה בגודל $N \times N$ תיוצג באמצעות $N \times N + 1$ מספרים. המספר הראשון יהיה גודל המטריצה, כלומר N . אחריו יופיעו איברי המטריצה שורה אחר שורה. ניתן להניח שכל המספרים הם מספרים שלמים, ושהמספר הראשון בקלט הוא גדול מ 0 (מייצג את גודל המטריצה). כמוכן, ניתן להניח שמבנה המטריצה תקין – כלומר, אם המספר הראשון הוא X כלשהו, אחריו יופיעו X^* מספרים שלמים.

שלב התרגיל מכיל חלק גדול מהקוד שנדרש עבור התוכנית. בניית המערך הדו מימדי על פי הקלט כבר מומש עבורכם. גם הקוד שאחראי על ההדפסות כבר נתון. מה שנדרש מכם הוא להשלים את החלק של סיבוב המטריצה ב 90 מעלות ימינה.

לדוגמא: עבור הקלט הבא לתוכנית:

3 1 2 3 4 5 6 7 8 9

הקוד הקיים בשלב יוצר מערך דו מימדי בגודל 3 על 3 (3 שורות ו3 עמודות)

[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

אחרי סיבוב של 90 מעלות (הקוד שלכם) תודפס המטריצה הבאה:

[7, 4, 1]
[8, 5, 2]
[9, 6, 3]

דוגמא נוספת: עבור הקלט:

4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

הקוד הקיים בשלב יוצר מערך דו מימדי בגודל 4 על 4 (4 שורות ו4 עמודות)

[1, 2, 3, 4]
[5, 6, 7, 8]
[9, 10, 11, 12]
[13, 14, 15, 16]

סיבוב של 90 מייצר את המטריצה הבאה:

[13, 9, 5, 1]
[14, 10, 6, 2]
[15, 11, 7, 3]
[16, 12, 8, 4]

אתגר - נסו להתייעל בשימוש בזיכרון. האם תוכלו לעשות זאת ללא יצירה של מערך נוסף מלבד המערך הדו-מימדי שמייצג את המטריצה? כלומר שהשינויים יהיו במטריצה הנתונה in place ובקוד שלכם לא יהיה שימוש ב new מעבר לקוד הנתון. זו לא חובה וניתן להגיש קוד שמשתמש בכל כמות זיכרון ומערכים שתמצאו.

בהצלחה!!