

# מבוא למדעי המחשב - סמסטר א' תשפ"ב

## עבודת בית מספר 1

### צוות העבודה:

- מרצה אחראית: מיכל שמש
- מתרגלים אחראים: רון שפירא ובר, אריאל חוזמן.

מועד פרסום: 24.10.21 בשעה 14:00.

מועד אחרון להגשה: 7.11.21 בשעה 23:59.

### 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 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. לעבודה זו מצורף מסמך הדרכה לבדיקה עצמית: **SelfTestingGuidelines.pdf**. קראו אותו בעיון.
3. עבודה זו תוגש ביחידים במערכת המודל ניתן לצפות בסרטון הדרכה על הגשת העבודה במערכת ה-vpl בלינק הבא: סרטון הדרכה.
4. במערכת מופיעים קבצי 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) וקובץ ה-readme.txt. אלו הם קבצי השלד אותם עליכם לערוך ולהגיש. עליכם לערוך את הקבצים האלו בהתאם למפורט בתרגיל ולהגישם כפתרון. **אין לשנות את שמות קבצי השלד.**
5. **המלצה על דרך העבודה** - אנו ממליצים לפתוח פרויקט ב-eclipse בשם Assignment1. כשתעבדו, תערכו (לאחר שהורדתם את קבצי השלד וחילצתם אותם לתוך הפרויקט) בתוך הפרויקט את הקבצים: Task1, Task2, Task3a, Task3b, Task4a, Task4b, Task4c, Task4d, Task4e, Task4f. בהתאם להוראות המשימה והגישו אותם לפי ההנחיות.

#### הנחיות נוספות


1. בכל קובץ מופיעה שורה המגדירה משתנה אשר יהווה את הפלט של התוכנית בינתן פלט כלשהו. לדוגמה:

```
int ans = 0
```

2. כמובן כי ייתכן וטיפוס המשתנה יהיה שונה כתלות בשאלה. בנוסף ייתכן ויהיו שני פלטים לתוכנית. במקרה כזה יוגדרו שני משתנים כמתואר בהמשך.
3. משתנה זה יהווה את פלט התוכנית, ובהוראות עבודה זו נתייחס אליו בשם זה.
4. עליכם להציב במשתנה זה את פלט התוכנית כך שבסוף ריצת התוכנית שלכם, המשתנה יכיל את ערך הפתרון.
5. שימו לב שבכל הקבצים המסופקים לכם עליכם לכתוב את הפתרון שלכם אך ורק בין שתי השורות המוגדרות ע"י ההערות:

```
// -----write your code BELOW this line only! -----
```

```
// -----write your code ABOVE this line only! -----
```

6. אין לשנות את השורות המסופקות לכם בקבצי השלד, למעט אתחול שונה (במידה ואתם חשים שיש צורך לכך) למשתנה המהווה את פלט התוכנית. אין לשנות שמות משתנים.
7. אין להדפיס למסך דברים נוספים חוץ משורת ההדפסה המסופקת לכם בקובץ. הדפסות נוספות יגררו הורדה בציון. כמו כן, אין לערוך את שורת ההדפסה.
8. עבודות שלא יעברו קומפילציה במערכת יקבלו את ה**ציון 0** ללא אפשרות לערער על כך. אחריותכם לוודא שהעבודה שאתם מגישים עוברת תהליך קומפילציה במערכת (ולא רק ב-eclipse). להזכירכם, תוכלו לבדוק זאת ע"י לחיצה על כפתור ה-Evaluate. 
9. עבודות הבית נבדקות גם באופן ידני וגם באופן אוטומטי. לכן, יש להקפיד על ההוראות ולבצע אותן במדויק.
10. סגנון כתיבת הקוד יבדק באופן ידני. יש להקפיד על כתיבת קוד יעיל, 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 מיותרות, 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.

### עזרה והנחיה

1. 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.
2. בתגבור השני של הסמסטר, 27.10.21-24.10.21 נפתור באופן מודרך את משימות 1, 2, 4 כמו כן, אתם יכולים להיעזר בפורום ולפנות בשאלות לחבריכם לכיתה. צוות הקורס עובר על השאלות ונותן מענה במקרה הצורך. שימו לב, **אין לפרסם פתרונות בפורום**.
3. בכל בעיה אישית הקשורה בעבודה (מילואים, אשפוז וכו'), אנא פנו אלינו דרך מערכת הפניות, כפי שמוסבר באתר הקורס.
4. אנחנו ממליצים בחום להעלות פתרון למערכת המודל לאחר כל סעיף שפתרתם. הבדיקה תתבצע על הגרסה האחרונה שהועלתה (בלבד!).

### הערות ספציפיות לעבודת בית זו

1. בעבודה זו 4 משימות ו-10 תתי-משימות וסך הנקודות המקסימלי הוא 100. שימו לב שמספר הנקודות לכל תת-משימה אחיד (10 נקודות למשימה) ואינו מצביע על קושי המשימה.
2. 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.
3. בעבודה זו, בתוכניות אותן אתם מגישים, כל המשתנים עבור מספרים שלמים צריכים להיות מטיפוס **int**.
4. בכל המשימות ניתן להניח כי הקלט תקין.

### 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 [בסילבוס הקורס](#) אנא עשו זאת כעת.

**משימות:**

יש להגיש את כל השאלות עד התאריך 7.11.21 תחת **עבודת בית 1 - VPL**. עקבו אחרי הוראות ההגשה בסוף העבודה.

**הצהרה (0 נקודות)**

פתחו את הקובץ readme.txt וכתבו בו את שמכם ומספר תעודת הזהות שלכם. משמעות פעולה זו היא שאתם מסכימים על הכתוב בו. דוגמה:

I, <Israel Israeli> (<123456789>), assert that the work I submitted is entirely my own.

I have not received any part from any other student in the class, nor did I give parts of it for use to others.

I realize that if my work is found to contain code that is not originally my own, a formal case will be opened against me with the BGU disciplinary committee.

**הקדמה: חילוק שלמים ושארית חלוקה**

לכל שני מספרים שלמים  $a, b$  כך ש-  $b \neq 0$ , החלק השלם במנה  $\frac{a}{b}$  הוא מספר שלם  $q$  כך ש-  $a = q \cdot b + r$  ו-  $r < b$ . המספר  $r$  נקרא שארית החלוקה של  $a$  ב-  $b$  ומסומנת  $r = a \% b$ .

למשל בעבור  $a = 13$  ו-  $b = 3$  החלק השלם במנה  $\frac{13}{3}$  הוא  $q = 4$  ושארית החלוקה  $r = 1$  כיוון שמתקיים  $13 = 4 \cdot 3 + 1$ .

**משימה 1 - משימת חימום**

פתחו את הקובץ Task1.java וכתבו בו תכנית אשר קולטת מהמשתמש ארבעה מספרים שלמים  $a, b, q, r$  ומציבה במשתנה **boolean ans** את הערך **true** אם  $r < b$ ,  $b \neq 0$ , ו-  $a = q \cdot b + r$  ו- **false** אחרת. המספרים ייקלטו בסדר הבא (משמאל לימין):  $a, b, q, r$ . בשאלה זו ניתן להניח כי הקלטים:  $a, b, q, r$  הם מספרים שלמים וכי  $b \geq 0$ . שימו לב, לא ניתן להניח דברים נוספים על הקלט.

דוגמאות:

אם הקלט הוא  $a = 10, b = 4, q = 2, r = 1$  אזי התוכנית תציב במשתנה **ans** את הערך: **false**

אם הקלט הוא  $a = 10, b = 0, q = 2, r = 2$  אזי התוכנית תציב במשתנה **ans** את הערך: **false**

אם הקלט הוא  $a = 9, b = 3, q = 3, r = 0$  אזי התוכנית תציב במשתנה **ans** את הערך: **true**

אם הקלט הוא  $a = 5, b = 7, q = 0, r = 5$  אזי התוכנית תציב במשתנה **ans** את הערך: **true**

בסוף ריצת התוכנית על המשתנה **ans** שסיפקנו לכם להכיל את הפתרון.

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

**משימה 2 – עוד משימת חימום**

פתחו את הקובץ Task2.java וכתבו בו תכנית אשר קולטת מהשתמש שני מספרים שלמים  $a, b$  כך ש  $a \leq b$  ומציבה במשתנה `int ans` מספר שלם  $n$  בתחום  $[a, b]$  אותו היא מגרילה באקראי. במילים: המספר  $n$  שיוגרל צריך לקיים  $a \leq n \leq b$ .

הדרכה: יש להשתמש בפקודה `Math.random()` המחזירה מספר אקראי  $x$  בתחום החצי פתוח  $[0, 1)$ . במילים:  $x$  שמוחזר ע"י הפקודה מקיים  $0 \leq x < 1$ . המספרים ייקלטו בסדר הבא (משמאל לימין):  $a, b$ . ניתן להניח כי הקלט תקין, כלומר כי  $a \leq b$ . הם מספרים שלמים וכן כי  $a \leq b$ .

דוגמאות:

אם הקלט הוא  $a = 2, b = 24$  אזי ערך אפשרי שיוצב במשתנה `ans` יכול להיות: 17

אם הקלט הוא  $a = -4, b = 5$  אזי ערך אפשרי שישתמש במשתנה `ans` הוא: -4

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

הערה – אתם לא חייבים ליצור משתנה נוסף  $n$ . ניתן להציב את התוצאה ישירות ב-`ans`. בסוף ריצת התוכנית על המשתנה `ans` שסיפקנו לכם להכיל את הפתרון.

**משימה 3: חזקות של 2 ושארית חלוקה****משימה 3א:**

פתחו את הקובץ Task3a וכתבו בו תכנית אשר קולטת מהשתמש מספר שלם אי-שלילי  $n$  ומציבה במשתנה `int ans` את הערך  $2^n$ . זיכרו כי יש להשתמש במשתנים מטיפוס `int` בלבד. על התוכנית לחשב נכונה את החזקות של 2 עבור כל ערך של  $n$  בין 0 ל-30 כולל.

דוגמאות:

אם הקלט הוא  $n = 0$  אז במשתנה `ans` יוצב הערך: 1

אם הקלט הוא  $n = 1$  אז במשתנה `ans` יוצב הערך: 2

אם הקלט הוא  $n = 10$  אז במשתנה `ans` יוצב הערך: 1024

אם הקלט הוא  $n = 31$  אז במשתנה `ans` יוצב הערך: -2147483648

נסו להבין מדוע.

**שימו לב:** בחלק זה אין להשתמש בספרייה `Math`. עליכם לחשב את  $2^n$  ע"י שימוש בלולאה.

ניתן להניח כי הקלט תקין, כלומר כי  $n$  הוא שלם אי-שלילי בין 0 ל-30 כולל. בסוף ריצת התוכנית על המשתנה `ans` שסיפקנו לכם להכיל את הפתרון.

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

**משימה 3:**

`int MV = Integer.MAX_VALUE;` מייצג את הערך המתקבל מהפקודה

פתחו את הקובץ Task3b.java וכתבו בו תכנית אשר קולטת מהמשתמש שני מספרים שלמים  $n, k$  ומציבה במשתנה `int ans` את הערך של  $2^n \% k$ , כלומר את שארית החלוקה של  $2^n$  ב- $k$ . המספרים ייקלטו בסדר הבא (משמאל לימין):  $n, k$ .

דוגמאות:

אם ערכי הקלט הם  $n = 10, k = 54$  אזי הערך שיוצב יהיה:

52

כיוון ש-  $2^{10} = 54 * 18 + 52$

אם בקלט שני הערכים הם  $n = 35, k = 151$  אזי הערך שיוצב יהיה:

32

כיוון ש-  $2^{35} = 151 * 227,547,936 + 32$

יש להניח כי המספרים  $n, k$  הם שלמים אי-שליליים וכי  $1 < k < \sqrt{MV}$ .

על התוכנית לחשב נכונה את  $2^n \% k$  לכל ערך כנ"ל של  $n$  ו- $k$  (בפרט עבור  $n \geq 31$ ).

הדרכת חובה: על מנת לפתור נכונה תרגיל זה גם עבור ערכים גדולים של  $n$ , יש להשתמש בעובדה הבאה:

$$(a \cdot b) \% k = ((a \% k) \cdot (b \% k)) \% k$$

לדוגמה:

$$(6 \cdot 7) \% 5 = 2 = ((6 \% 5) \cdot (7 \% 5)) \% 5$$

בסוף ריצת התוכנית על המשתנה `ans` שסיפקנו לכם להכיל את הפתרון.

**משימה 4: במשימה זו נדון בבעיית בדיקת ראשוניות של מספר**

להזכירכם: `int MV = Integer.MAX_VALUE;` מייצג את הערך המתקבל מהפקודה

**משימה 4א: אלגוריתם נאיבי לבדיקת ראשוניות של מספר**

תזכורת:

מספר ראשוני (prime)  $p$  הוא מספר שלם גדול מ-1 אשר מתחלק ללא שארית רק ב-1 ובעצמו. לדוגמה: 2,3,5,7,...  
מספר פריק (composite) הוא מספר שלם אשר קיים לו מחלק שלם גדול מ-1 השונה מ-1 ומעצמו. לדוגמה: 4,6,8,9,...  
הנחה: בכל חלקי המשימה הבאים אין להשתמש בפונקציה `Math.pow` (למעט עבור בדיקות נכונות אשר תכתבו בעצמכם ואינן כלולות בקוד המוגש).

פתחו את הקובץ Task4a.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם  $n$  כך ש-  $1 < n \leq MV$  ומציבה במשתנה `boolean ans` את הערך `true` אם  $n$  ראשוני ו-`false` אחרת. הדרכת חובה: יש לבדוק בלולאה האם קיים ל- $n$  מחלק שאינו טריוויאלי.

דוגמאות:

אם הקלט הוא  $n = 10$  אזי הערך במשתנה `ans` יהיה:

false

אם הקלט הוא  $n = 11$  אזי הערך במשתנה `ans` יהיה:

true

ניתן להניח כי הקלט תקין, כלומר כי  $1 < n \leq MV$ .

בסוף ריצת התוכנית על המשתנה `ans` שסיפקנו לכם להכיל את הפתרון.

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

**משימה 4ב: אלגוריתם נאיבי לבדיקת מספר ראשוניים**מספר הראשוניים

לכל מספר שלם  $n > 1$  נסמן ב-  $\pi(n)$  את מספר המספרים הראשוניים אשר קטנים או שווים ל-  $n$ .  
לדוגמה:  $\pi(2) = 1, \pi(5) = 3, \pi(20) = 8$

פתחו את הקובץ Task4b.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר שלם  $n$  ומחשבת את  $\pi(n)$ .  
על הפתרון להיות מוכל במשתנה `int ans` שסיפקנו לכם.

דוגמאות:

אם הקלט הוא  $n = -10$  אז הערך שיוכל ב `ans` יהיה:  
0

אם הקלט הוא  $n = 0$  אז הערך שיוכל ב `ans` יהיה:  
0

אם הקלט הוא  $n = 2$  אז הערך שיוכל ב `ans` יהיה:  
1

אם הקלט הוא  $n = 5$  אז הערך שיוכל ב `ans` יהיה:  
3

אם הקלט הוא  $n = 20$  אז הערך שיוכל ב `ans` יהיה:  
8

סיימתם חלק זה? כל  
הכבוד! העלו את הגרסה  
האחרונה של עבודתם  
למערכת המודל.

ניתן להניח כי הקלט תקין, כלומר כי  $n$  הוא מספר שלם ו-  $n \leq MV$ .  
בסוף ריצת התוכנית על המשתנה `ans` שסיפקנו לכם להכיל את הפתרון.

אלגוריתם מילר-רבין – מבוא

האלגוריתם הנאיבי לבדיקת ראשוניות מספר נתון מסוגל לספק תשובה נכונה בזמן סביר עבור מספרים שאינם מאוד גדולים. כאשר מדובר במספרים גדולים (בעלי 200 ספרות, לדוגמה) האלגוריתם הנאיבי ירוץ בזמן ארוך מדי ולא נוכל לקבל תשובה בזמן סביר.

אחד האלגוריתמים הנפוצים והמקובלים כיום לבדיקת ראשוניות של מספר גדול הוא האלגוריתם של מילר-רבין.  
אלגוריתם זה הוא אלגוריתם אקראי, דבר הגורר אפשרות שתוחזר תשובה שגויה (בהמשך ישנו פירוט לגבי הסיכוי לשגיאה שלו).

בתרגיל מודרך זה (בו כל סעיף מסתמך על סעיפים קודמים) נבדוק האם מספר נתון הוא ראשוני. הבדיקה תיעשה בהתאם לאלגוריתם מילר-רבין.

ראשית (משימות ג'-ה') נקלוט מהמשתמש מספר אי-זוגי  $n$  ונבצע בדיקה בודדת עבור ראשוניות המספר  $n$  שהגרלנו עם הסתברות להחזרת תשובה שגויה שלא עולה על  $\frac{1}{4}$ . בבדיקה זו נבדוק האם מספר  $b$  בתחום  $[2, n-1]$  שהוגרל באקראי עומד בתנאי מסויים שהוגדר ע"י מילר-רבין (אותו נציג בהמשך) ביחס ל-  $n$ , נסמנו (\*). למספר  $b$  בתחום  $[2, n-1]$  אשר עומד בתנאי (\*) עבור  $n$  נקרא עד (witness), כיוון שהוא מעיד על כך ש-  $n$  מספר פריק.

לבסוף (משימה ו') נחזור על הבדיקה באופן בלתי תלוי  $k$  פעמים ובכך נקטין את ההסתברות לשגיאה כך שלא תעלה על

$$\left(\frac{1}{4}\right)^k. \text{ נשים לב כי עבור הערך } k = 50 \text{ אנו מקטינים את ההסתברות לשגיאה ל- } \frac{1}{2^{100}} = \frac{1}{4^{50}}.$$

סדר הפעולות (האלגוריתם) בשלב הראשון יהיה כדלהלן:

1. נקלוט מהמשתמש מספר אי-זוגי  $n > 1$ .
2. נגריל מספר שלם יחיד  $b$  בתחום  $[2, n - 1]$ .
3. נבדוק האם  $b$  שהגרלנו עומד בתנאי (\*) ביחס ל- $n$ .
4. אם התנאי (\*) מתקיים, נכריז כי  $n$  פריק.
5. אחרת, נכריז כי  $n$  ראשוני.

עובדות מתמטיות:

- ❖ אם  $n$  פריק, ישנם לפחות  $\frac{3}{4}n$  מספרים  $b$  בתחום  $[2, n - 1]$  אשר יעמדו בתנאי (\*) ולכן ההסתברות להגריל באקראי  $b$  העומד בתנאי גדולה או שווה ל- $\frac{3}{4}$ .
- ❖ אם  $n$  פריק, ישנם פחות מ- $\frac{1}{4}n$  מספרים  $b$  בתחום  $[2, n - 1]$  אשר לא יעמדו בתנאי (\*) ולכן ההסתברות להגריל באקראי  $b$  שלא עומד בתנאי קטנה מ- $\frac{1}{4}$ .
- ❖ אם  $n$  ראשוני אז כל  $b$  בתחום  $[2, n - 1]$  לא עומד בתנאי (\*).

לאור עובדות אלו נוכל להבין את האלגוריתם באופן הבא:

1. אם הכרזנו כי  $n$  פריק, אזי אנו יודעים בוודאות כי הוא פריק.
2. אם הכרזנו כי  $n$  ראשוני, קיימים שני מצבים אפשריים:
  - $n$  ראשוני. אז החזרנו תשובה נכונה.
  - $n$  פריק. אז טעינו: מכיוון שהוגרל מספר  $b$  אשר לא עמד בתנאי (\*).

נצא לדרך!

**משימה 4:**

**להזכירכם:**  $MV$  מייצג את הערך המתקבל מהפקודה `int MV = Integer.MAX_VALUE;` פתחו את הקובץ Task4c.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר אי זוגי  $n$  כך ש- $1 < n < \sqrt{MV}$ , מגרילה מספר שלם  $b$  בתחום  $[2, n - 1]$  ומציבה אותו במשתנה `int ans`. ניתן להניח כי הקלט תקין, כלומר כי  $n$  הוא שלם אי זוגי כך ש- $1 < n < \sqrt{MV}$ .

דוגמא:

אם הקלט הוא  $n = 22,317$  אזי ערך אפשרי שיוצב במשתנה `ans` הוא 1684

בסוף ריצת התוכנית על המשתנה `ans` שסיפקנו לכם להכיל את הפתרון.

**משימה 74:**

בהינתן מספר  $x > 1$  זוגי ניתן לייצגו באופן הבא  $x = 2^s \cdot d$  כאשר  $s > 0$  ו- $d$  אי-זוגי.  
דוגמאות:

$$6 = 2^1 \cdot 3$$

$$60 = 2^2 \cdot 15$$

פתחו את הקובץ Task4d.java וכתבו בו תכנית אשר קולטת מהמשתמש מספר אי-זוגי  $n$  כך ש- $1 < n < \sqrt{MV}$  ומוצאת את  $s, d$  כך ש- $n - 1 = 2^s \cdot d$ ,  $s > 0$  ו- $d$  אי-זוגי. התוכנית תציב אותם במשתנים `int ans1` ו-`int ans2`, באופן הבא:

המספר שיוצב במשתנה `ans1` יהיה  $s$ . המספר שיוצב במשתנה `ans2` יהיה  $d$ .

ניתן להניח כי הקלט תקין, כלומר כי  $n$  הוא שלם אי-זוגי כך ש- $1 < n < \sqrt{MV}$ .  
הדרכת חובה: את החישוב של  $s$  בצעו ע"י שימוש בלולאה.  
דוגמא:

בהנחה והקלט הוא המספר 12,317 הערכים שיוצבו יהיו:

2 במשתנה `ans1`

3079 במשתנה `ans2`

כיוון ש- $12,317 - 1 = 2^2 \cdot 3,079$ .

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

בסוף ריצת התוכנית על המשתנים `ans1` ו-`ans2` שסיפקנו לכם להכיל את הפתרון.

סימון: שקילות מודולו  $n$ .

יהיו  $n > 1$  מספר טבעי ו- $a$  מספר טבעי. אם שארית החלוקה של  $a$  ב- $n$  היא  $b$  (כלומר  $a \% n = b$ ) אז נסמן כי  $a \equiv b \pmod{n}$  (נאמר ש- $a$  ו- $b$  שקולים מודולו  $n$ ).

דוגמאות:  $4 \equiv 1 \pmod{3}$ ,  $13 \equiv 6 \pmod{7}$

באופן כללי יותר:

יהיו  $n > 1$  מספר טבעי ו- $c, d$  מספרים טבעיים.

נאמר כי  $c$  ו- $d$  שקולים מודולו  $n$  ונסמן  $c \equiv d \pmod{n}$  או  $d \equiv c \pmod{n}$  אם מתקיים כי  $c \% n = d \% n$ .



**משימה 4:**

פתחו את הקובץ Task4e.java וכתבו בו תכנית אשר קולטת מהמשתמש 4 מספרים  $n, b, s, d$  בסדר הזה משמאל לימין כך ש:

- $n$  מספר אי-זוגי בתחום  $[2, \sqrt{MV}]$ .
  - $b$  מספר שלם בתחום  $[2, n - 1]$ .
  - $s, d$  שלמים כך ש-  $n - 1 = 2^s \cdot d$ ,  $s > 0$ , ו-  $d$  אי-זוגי.
- ובודקת האם  $b$  עומד בתנאי הבא, אשר נסמנו ב- (\*) עבור  $n$ :

תנאי (\*) מציין כי:

1. לא מתקיים  $b^d \equiv 1 \pmod{n}$
  - וגם
  2. לכל  $0 \leq i \leq s - 1$  לא מתקיים:  $b^{2^i \cdot d} \equiv (n - 1) \pmod{n}$
- כלומר, כל  $s + 1$  השקילויות לעיל לא מתקיימות.

אם התנאי מתקיים התוכנית תציב במשתנה `boolean ans1` את הערך `false` ובמשתנה `int ans2` את העד שמצאתם (b) אחרת התוכנית תציב במשתנה `boolean ans1` את הערך `true` ובמשתנה `int ans2` את הערך -1. ניתן להניח כי הקלט תקין, כלומר כי:

- $n$  מספר אי-זוגי בתחום  $[2, \sqrt{MV}]$ .
- $b$  מספר שלם בתחום  $[2, n - 1]$ .
- $s, d$  שלמים כך ש-  $n - 1 = 2^s \cdot d$ ,  $s > 0$ , ו-  $d$  אי-זוגי.

דוגמאות:

אם ערכי הקלט הם  $n = 25,123$ ,  $b = 11,309$ ,  $s = 1$ ,  $d = 12,561$  אזי: הערך שיוצב במשתנה `ans1` יהיה `false` והערך שיוצב במשתנה `ans2` יהיה: 11309 במילים אחרות, מצאנו עד (b) לכך ש-  $n$  הוא פריק. זאת מפני ש-  $b$  עמד בתנאי (\*).

אם ערכי הקלט הם  $n = 31,663$ ,  $b = 16,116$ ,  $s = 1$ ,  $d = 15,831$  אזי: הערך שיוצב במשתנה `ans1` יהיה: `true` והערך שיוצב במשתנה `ans2` יהיה: -1

(זאת כיוון ש-  $16,116^{15,831} = 31,662 \pmod{31,663}$ )

במילים אחרות,  $b$  לא עמד בתנאי (\*). לכן נאמר ש-  $n$  הוא ראשוני עם הסתברות שגיאה של פחות מ-  $\frac{1}{4}$ .

בשתי הדוגמאות לא ניתן לייצג במשתנה מסוג `int` את הערך  $b^{2^i \cdot d}$  אפילו עבור ערכי  $i$  קטנים. על הפתרון שלכם לעבוד גם עבור מצבים אלו.

בסוף ריצת התוכנית על המשתנים `ans1` ו- `ans2` שסיפקנו לכם להכיל את הפתרון. רמז: ראו הדרכת חובה במשימה 3ב.

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.

**משימה 14:**

פתחו את הקובץ Task4f.java וכתבו בו תכנית אשר קולטת מהמשתמש 4 מספרים  $n, s, d, k$  בסדר הזה משמאל לימין כך ש-

- $n$  מספר אי-זוגי בתחום  $[2, \sqrt{MV}]$ .
- $s, d$  שלמים כך ש-  $n - 1 = 2^s \cdot d$ ,  $s > 0$ , ו-  $d$  אי-זוגי.
- $k$  שלם כך ש-  $1 < k < 51$ .

ופועלת באופן הבא:

התוכנית חוזרת על הפעולה הבאה  $k$  פעמים:

- התוכנית מגרילה מספר  $b$  שלם בתחום  $[2, n - 1]$  ובודקת האם  $b$  עומד בתנאי (\*) עבור  $n$ .
- אם באחת ההגרלות הוגרל מספר  $b$  אשר עמד בתנאי (\*) תציב במשתנה **boolean ans** את הערך false אשר מסמל כי  $n$  פריק.
- אחרת, התוכנית תציב במשתנה **boolean ans** את הערך true אשר מסמל כי  $n$  ראשוני.

שימו לב: אין חובה לבצע את כל  $k$  ההגרלות במידה והוגרל מספר  $b$  אשר עמד בתנאי (\*) עבור  $n$ . ניתן להניח כי הקלט תקין, כלומר כי:

- $n$  מספר אי-זוגי בתחום  $[2, \sqrt{MV}]$ .
- $s, d$  שלמים כך ש-  $n - 1 = 2^s \cdot d$ ,  $s > 0$ , ו-  $d$  אי-זוגי.
- $k$  שלם כך ש-  $1 < k < 51$ .

דוגמאות:

סיימתם חלק זה? כל הכבוד! העלו את הגרסה האחרונה של עבודתם למערכת המודל.





אם ערכי הקלט הם  $n = 4,793, s = 3, d = 599, k = 10$  אזי הערך שיוצב במשתנה **ans** יהיה: **true**  
 אם ערכי הקלט הם  $n = 15,379, s = 1, d = 7689, k = 15$  אזי הערך שיוצב במשתנה **ans** יהיה: **false** (הסיכוי לשגיאה הוא אפסי. בערך 1 ל-1,073,741,824).

הערה: בתוכנית זו, ההסתברות לשגיאה אינה עולה על  $\left(\frac{1}{4}\right)^k$ .

במילים אחרות: אם נקלט מספר  $n$  פריק, ההסתברות שנצהיר כי הוא ראשוני לא עולה על  $\left(\frac{1}{4}\right)^k$ .

אם נקלט מספר  $n$  ראשוני, כל  $b$  שיוגרל לא יעמוד בתנאי (\*) עבר  $n$  ולכן נצהיר בוודאות כי הוא ראשוני. בסוף ריצת התוכנית על המשתנה **ans** שסיפקנו לכם להכיל את הפתרון.

**הוראות הגשה:**

1. גשו ל-**עבודת בית 1 – VPL** באתר הקורס.
2. גשו ללשונית Edit.
3. לחצו על הכפתור ה-.
4. יפתחו לכם עוד אופציות, בין היתר אופציה של  upload לחצו על הכפתור ובחרו את הקבצים שערכתם בפרויקט Assignment1. **ודאו כי לא חסרים קבצים וכי הקבצים שהעליתם הם הקבצים המעודכנים ביותר.**
5. שמרו את השינויים (יש ללחוץ על כפתור השמירה) .
6. לחצו על Evaluate .
7. אתם אמורים לקבל פידבק עבור הצלחתכם בבדיקות החלקיות שרצות בזמן הגשה זו (בדיקות נוספות יתבצעו בתום תאריך ההגשה).
8. אנו חוזרים ואומרים, זו אחריותכם לבדוק שהקבצים שהגשתם עוברים תהליך קומפילציה במערכת. עבודות שלא יתקמפלו יקבלו את הציון 0.

סטודנטים המתעניינים בקריאה נוספת מוזמנים לקרוא על המושגים הבאים (בהם תיתקלו במהלך לימודיכם):

1. מספר ראשוני – Prime Number
2. פירוק מספר לגורמים ראשוניים – Finding Factors of a Number
3. חשבון מודולרי (חשבון קונגראנציות) – Modular Arithmetic
4. המשפט הקטן של פרמה – Fermat's Little Theorem
5. אלגוריתם דטרמיניסטי – Deterministic Algorithm
6. אלגוריתם אקראי – Randomized Algorithm
7. צפיפות הראשוניים

# בהצלחה !