

מטלה 3- מבוא לחישוב. סמסטר ב 2021

הנחיות כלליות:

- תרגיל זה נעשה ביחידים בלבד.
- הקפידו לבדוק ולטפל בכל סוגי הקלטים, מצורפת דוגמת הרצה של התרגיל, אשר מדגימה קליטים \ פלטים שונים.
- לכל סעיף יש לפתוח קובץ חדש בשם: Ex3, מקף תחתון, מספר תרגיל, מקף תחתון מספר סעיף. לדוגמה, הקובץ שפותר את התרגיל 1 סעיף 1 יקרא בשם Ex3_1_1, הקובץ שפותר את התרגיל 1 סעיף 2 יקרא בשם Ex3_1_2, הקובץ שפותר את התרגיל 2 שאין בו סעיפים יקרא בשם Ex3_2, שמות מדויקות נמצאות בסוף המסמך.
- יש להגיש קובץ zip. מספר_זהות שבתוכו יש לשים את כל הקבצים שיצרתם אותם. (לא יתקבלו עבודות שנשלחו בדואר אלקטרוני!)
- כתובת ההגשה של התרגיל: מודל.
- אין צורך לצרף קבצים StdDraw.java ו- MyConsole.java
- יש להשתמש אך ורק בשמות שהוגדרו במטלה.

חלק ראשון: משחק – ניחוש

את הפונקציות בחלק זה יש לכתוב במחלקה EX3_1.java

תארו את המשחק הבא: סטודנט חושב על מספר $0 \leq m \leq N$ בין 0 ל-N ולתכנית יש לנחש

המספר של הסטודנט

במספר מינימאלי של הצעדים.

התוכנית מדפיסה מספר בין 0 ל-N ומבקשת את המשתמש לבחור באפשרות המתאימה:

(1) זה המספר שחשבת עליו

(2) האם המספר שחשבת עליו קטן ממספר שהדפסתי?

(3) האם המספר שחשבת עליו גדול ממספר שהדפסתי?

התשובה של המשתמש היא: < 1 או 2 או 3 >

הסטודנט בוחר באפשרות המתאימה ועונה לתוכנית.

המשחק נמשך עד שהסטודנט יקבל את המספר שלו.

במידה והמספר זהה למספר של הסטודנט (הסטודנט מקליד 1) התוכנית מדפיסה: "WIN!"

והמשחק מסתיים. **בסוף המשחק התוכנית מדפיסה את מספר הצעדים שהתבצעו עד ניחוש המספר.**

יש לכתוב פונקציה סטטית שמקבלת מספר שלם חיובי N – ומנחשת את המספר של הסטודנט:

```
public static void geussNumber(int N)
```

הערה: כאשר נקלט מספר שונה מ-1,2,3 יש לבקש הקלטה נוספת, עד שנקלט מספר נכון: 1,2, או 3.

חלק שני: רקורסיות ומחרוזות:

א. נוסחה היא מחרוזת המכילה מספרים **חיוביים**, **חד-ספרתיים**, משתנה ופעולות (+, -, *, /). המשתנה יהיה מסומן תמיד כ- x (לא מקדמים). המחרוזת לא מכילה רווחים.
כל פעולה בין 2 מספרים (או משתנים) מופרדת בסוגריים. **לדוגמא:** $(5+((3+x)*(4-5)))$

1. כתבו פונקציה **רקורסיבית** בוליאנית, המקבלת מחרוזת s ומחזירה $true$ אם היא תקנית ו- $false$ אחרת.

לדוגמא:

ביטויים **תקינים**, הפונקציה תחזיר **true**:

$((1+x)*(2+9))$, $((7*x)+(3+x))*(4-5)$
1,
 $(5+4)$,
 $(4*((4\%3)+2))$,
 $((3+1)*(9-2))$

ביטויים **לא** תקינים, הפונקציה תחזיר **false**:

$(4+*(3+x)*(4-5))$
 $((4+6+x)*(4-5))$
-1,
4+3,
 $(21+3)$,
78,
 $((2+1))$

חתימת הפונקציה:

```
public static boolean isFormula(String s)
```

2. כתבו פונקציה **רקורסיבית** המקבלת מחרוזת s ומספר x ומציבה את המספר בנוסחה אם היא תקינה. **לדוגמא:** עבור $x=2$, $s=(5+((3+x)*(4-5)))$. כאשר הביטוי במחרוזת אינו תקין הפונקציה מחזירה **NaN**. **Double.NaN**.
In **Java**, "**NaN**" stands for "**not a number**" and signifies that a value is **not** defined.

חתימת הפונקציה:

```
public static double assign(String s, int x)
```

הערה חשובה: ניתן להשתמש בפונקציית עזר שמחשבת את הפעולה המרכזית ומכילה לולאה.

פתרון לא רקורסיבי או פתרון שמכיל רקורסיה בתוך לולאה לא יזכה בניקוד מלא.

ב. מספר ניתן לפירוק 3 אם ניתן להציגו כחיבור של חזקות 3.

לדוגמא: 91,21 ניתנים לפירוק 3 כי:

$$3^0 + 3^2 + 3^4 = 91$$

$$3^1 + 3^2 = 12$$

אבל 14- לא ניתן לפירוק 3

כתבו פונקציה (לא רקורסיבית) המקבלת מספר ומחזירה true אם הוא ניתן לפירוק 3 ו false אחרת.

אם המספר הוא פריק יש להדפיס את הפירוק שלו.

חתימת הפונקציה:

```
public static boolean checkPowerThree(int x)
```

חלק שלישי: מיונים וחיפוש בינארי.

את הפונקציות בחלק זה יש לכתוב במחלקה EX3.java

א.

1. כתבו פונקציה המקבלת שני מספרים m, n ומחזירה את שארית החלוקה של m ב- n (מבלי להשתמש ב %).
חתימת הפונקציה:

```
public static int mod(int m,int n)
```

2. כתבו פונקציה המקבלת מערך של מספרים שלמים: a ומספר n . הפונקציה תמייין את המערך לפי שארית החלוקה ב n . לדוגמא: עבור הקלט: $a=[3,8,9,5]$, $n=7$ הפונקציה תחזיר: $[8,9,3,5]$, כי:
 $8\%7=1$, $9\%7=2$, $3\%7=3$, $5\%7=5$.
(יש להשתמש באחד מהמיונים שלמדנו בכיתה)
חתימת הפונקציה:

```
public static int[] sortModArray(int[] a,n)
```

3. כתבו פונקציה שמקבלת מערך ממין a ומספר n כפי שהוגדר בסעיף הקודם. הפונקציה מקבלת מספר num ומחפשת אותו במערך, זמן החיפוש צריך להיות לוגריתמי (יש להשתמש בחיפוש בינארי). אם המספר אינו קיים יש להחזיר -1.
לדוגמא:

עבור המערך: $a=[8,9,3,5]$, $n=7$, $num=9$. הפונקציה תחזיר: 1
עבור המערך: $a=[8,9,3,5]$, $n=7$, $num=2$. הפונקציה תחזיר: -1

חתימת הפונקציה:

```
public static int searchModArray(int[] a,n, num)
```

* הפונקציה מחזירה את מקום האיבר במערך או 1-

ב.

4. חישוב פונקציית אוילר.

הגדרה: מספר a נקרה ראשוני ביחס ל- b אם המחלק המשותף הגדול ביותר של a ו- b שווה 1, כלומר $\gcd(a,b)=1$. לדוגמא, מספר 24 הוא ראשוני ביחס ל-5, כי המחלק המשותף הגדול שלהם הוא 1.

פונקציית אוילר (Euler): φ היא פונקציה מאוד חשובה בתורת המספרים. פונקציה $\varphi(n)$ מוגדרת כמספר שלמים חיוביים (כולל 1) קטנים מ- n שהם ראשוניים ביחס ל- n .

דוגמא 1: $\varphi(5)=4$

בגלל שהמחלק המשותף הגדול ביותר של 5 וכל אחד ממספרים 1,2,3,4 שווה 1.

דוגמא 2: $\varphi(6)=2$

בגלל שבמספרים מ-1 עד 5, רק $\gcd(6,1)=1$ ו- $\gcd(6,5)=1$.

דוגמא 3: $\varphi(1)=1$

דוגמא 4: $\varphi(2)=1$

כתבו **פונקציה** שמקבלת מספר שלם חיובי גדול או שווה 1 ומחזירה ערך של פונקציית אוילר:

```
public static int phi(int n)
```

בהצלחה רבה!!

נספח – שמות הקבצים.

חלק ראשון

```
public class Ex3_1 {  
    public static void geussNumber(int N) {...}  
}
```

חלק שני

```
public class Ex3_2_1 {  
    public static boolean isFormula(String s) {...}  
    public static double assign(String s, int x) {...}  
}
```

```
public class Ex3_2_2 {  
    public static boolean checkPowerThree(int x){...}  
}
```

חלק שלישי

```
public class Ex3_3_1 {  
    public static int mod(int m,int n){...}  
    public static int[] sortModArray(int[] a,n) {...}  
    public static int searchModArray(int[] a,n, num) {...}  
}
```

```
public class Ex3_3_2 {  
    public static int phi(int n) {...}  
}
```