

מבוא לחישוב – סמסטר ב' תש"פ מטלה מס' 2

הנחיות כלליות:

1. מטלה זאת נעשית **ביחידים** בלבד.
2. יש **לכתוב הערות** במקומות המתאימים ולהשתמש בשמות משתנים משמעותיים.
3. אחרי סיום המטלה, יש לדחוס את כל הקבצים עם סיומת **java** לתוך קובץ מסוג **zip** ששמו הוא **מס' תעודת הזהות** של הסטודנט.
4. המטלה תיבדק באמצעות בדיקות אוטומטיות. לכן, יש להקפיד על הפלטים בדיוק כפי שהוגדר.
5. הקפידו לבדוק ולטפל בכל סוגי הקלטים, מצורפת דוגמת הרצה של התרגיל, אשר מדגימה קלטים \ פלטים שונים.
6. **לא יתקבלו עבודות שנשלחו בדואר אלקטרוני!**
7. **כל הפונקציות יהיו תחת מחלקה אחת בשם M_2 ובתיקה בשם : M_2**

חלק ראשון:

בחלק זה עליכם לכתוב פונקציה שמחשבת את כל הגורמים (המחלקים) הראשוניים של מספר נתון. בתורת המספרים, **מספר ראשוני** הוא מספר טבעי גדול מ-1, שלא ניתן להציגו כמכפלה של שני מספרים טבעיים קטנים ממנו, כלומר הוא מתחלק רק ב-1 ובעצמו. מספר טבעי גדול מ-1 שאינו ראשוני נקרא **מספר פריק**. המספר 1 אינו נחשב ראשוני, וגם לא פריק. **גורמים הראשוניים** של מספר הם כל המספרים הראשוניים שכאשר מכפלים אותם מקבלים את המספר המקורי.

דוגמה:

גורמים ראשוניים של מספר 24 הם 2, 2, 2, 3 : $24 = 3 \cdot 2 \cdot 2 \cdot 2$
גורמים ראשוניים של מספר 117 הם 3, 3, 13 : $117 = 13 \cdot 3 \cdot 3$
גורמים ראשוניים של מספר 25 הם 5, 5 : $25 = 5 \cdot 5$

עליך לכתוב פונקציה

```
public static int[] primeFactors(int n){....}
```

קלט: n – מספר טבעי, $n \geq 2$.

פלט: מערך שמכיל את כל המחלקים הראשוניים של n.

אורך המערך צריך להיות שווה למספר מחלקים ראשוניים של n.

חלק שני:

בחלק זה עליכם לכתוב פונקציות שמחשבת סדרת מספרי המינג (Hamming numbers).

מספר המינג הוא מספר חיובי שצורתו היא $2^i \cdot 3^j \cdot 5^k$ עבור מספרים שלמים לא שליליים i, j, k .

מספר המינג קטן ביותר הוא 1: $1 = 2^0 \cdot 3^0 \cdot 5^0$,
מספר המינג שני בגודלו הוא 2: $2 = 2^1 \cdot 3^0 \cdot 5^0$

עליך לכתוב שתי פונקציות:
(א)

```
public static boolean isHammingNumber(int k) {...}
```

קלט: מספר טבעי $k \geq 1$

פלט: הפונקציה מחזירה true אם k הוא מספר המינג, אחרת היא מחזירה false.
(ב)

```
public static int[] hammingNumbers(int n) {.....}
```

קלט: מספר טבעי $n \geq 1$.

פלט: הפונקציה מחזירה מערך של n מספרי המינג.

לבדיקה: פלט עבור $n = 33$:

1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20,
24, 25, 27, 30, 32, 36, 40, 45, 48, 50, 54, 60,
64, 72, 75, 80, 81, 90, 96, 100.

חלק שלישי: העברת בסיסים בין עשרוני ובינארי

כידוע לכם, המחשב פועל בעזרת מספרים בבסיס בינארי (בעלי ערכים של 1 או 0) ואילו בני האדם סופרים בעזרת השיטה העשרונית. כאשר סופרים בשיטה בינארית המספר הראשון הוא 0, השני הוא 1, השלישי הוא 10, הרביעי הוא 11, החמישי הוא 101 וכו'.... האלגוריתם להעברת מספר בבסיס עשרוני לייצוגו הבינארי פועל בצורה הבאה:

לדוגמא: העברה של המספר העשרוני 22 לבסיסו הבינארי מתבצעת בצורה הבאה:

שארית	תוצאה	חילוק בסיס
0	11	22 / 2
1	5	11/2
1	2	5/2
0	1	2/2
1	0	1/2

החלוקה מתבצעת עד אשר הגענו לתוצאה של 0 במספר המחולק והתוצאה הבינארית מופיע בטרור השארית כאשר סדר הספרות הוא מלמטה למעלה. 22 בבסיס 10 = 10110 בבסיס 2. העברה מבסיס בינארי לעשרוני מתבצעת לפי הנוסחה הבאה:

$$101_2 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 5_{10}$$

$$10011001_2 = 1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 153_{10}$$

כתבו את אוסף הפונקציות הסטטיות הבאות:
(א)

```
public static int binary2int(int[] b){ . . . }
```

קלט: הפונקציה מקבלת מערך המורכב מאפסים ואחדות המייצג את מספר שלם גדול או שווה לאפס בצורה הבינארית.
פלט: הפונקציה מחזירה את המספר המיוצג (יש להניח קלט תקין).

(ב)

```
public static []int int2Binary (int n)
```

קלט: הפונקציה מקבלת מספר שלם n גדול או שווה לאפס.
פלט: הפונקציה מחזירה מערך המייצג את המספר בצורה הבינארית.

לדוגמא: `binary2int(b) b[]={1,0,0}` תחזיר 4
`int2Binary(153)` תחזיר {1,0,0,1,1,0,0,1}
`binary2int(int2binary(15))` תחזיר 15

ובכלל לכל n טבעי השוויון `n == Binary2int(int2Binary(n))` צריך להתקיים.

(ג)

```
public static boolean isBinaryArray(int[] b){...}
```

קלט: הפונקציה מקבלת מערך המורכב מספרים שלמים חד-ספרתיים (0,1,2,...,9) גדולים או שווים לאפס.

פלט: הפונקציה מחזירה true אם ורק אם המערך מורכב מאפסים ואחדות בלבד, אחרת היא מחזירה false.

לדוגמה:

```
isBinaryString(b), b[]={1,0,1,1} תחזיר true.  
isBinaryString(b), b[]={1,0,2,1} תחזיר false.
```

הערות:

כתבו תוכנית main (לא תיבדק) לבדיקה עצמית של כל הפונקציות שכתבתם. אחרי שסיימתם את העבודה בדקו את עצמכם שוב בעזרת תוכנית הבדיקה האוטומטית. על ידי כך תוודאו שהתוכנית שלכם כוללת את חתימות הפונקציות הנכונות – שימו לב התוכנית שלכם תיבדק גם על-ידי תוכנית אחרת.

צבירה מהנה