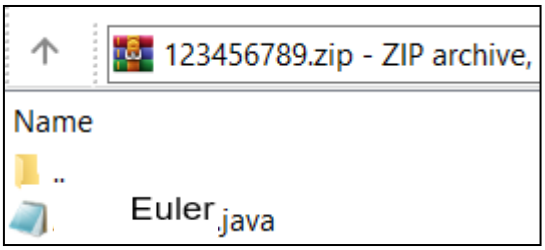
אלגוריתמים 2 סמסטר ב' התשפ"א – תחרות 3

מציאת מסלול אוילר

הנחיות:

- שפת תכנות – JAVA
- הפונקציות שעליכם לכתוב צריכות להיות קודם כל נכונות, ולאחר מכן **מהירות ככל האפשר**.
- תרגיל זה נעשה ביחידים בלבד.
- מועד אחרון להגשת המטלה: **26.07.21 - 23:58**.
- תרגיל זה ייבדק בצורה אוטומטית ע"י תוכנית מחשב שתשתמש בשמות הקבצים והפונקציות המוזכרים להלן.
- את התרגיל יש להגיש למקום המתאים במידע האישי שלכם.
- הקובץ **לא יהיה מוגדר תחת package!**
- שם הקובץ: Euler.java
- הקובץ הזה **לבדו** יהיה בתוך קובץ zip, ששמו תעודת הזהות שלכם. לדוגמא: 123456789.zip



- בתחילת הקובץ יש לרשום הערה עם מספר תעודת הזהות שלכם. לדוגמא:

```
// ID: 123456789
public class Euler {

}
```

- אופן הבדיקה: בשלב הראשון הקוד שלכם נבדק בצורה אוטומטית על אוסף טסטים. רק הסטודנטים שעוברים את **כל** הטסטים, עוברים לשלב השני : בדיקה מהירות. כאן, מי שמהיר יותר יזכה בניקוד גבוה יותר. כדי להיות מהיר ביותר, תוכלו להשתמש בכל אלגוריתם ומבנה נתונים שלמדנו \ נמצא בספרות \ באינטרנט \ לפתח בעצמכם.
אתם מוזמנים לחפש ולבחון את יעילות מבני הנתונים והאלגוריתמים השונים האפשריים.

קובץ שלא יוגש לפי הדרישות הנ"ל לא ייבדק! – אין ויתורים

ניסוח הבעיה:

עליכם לבדוק האם יש בגרף לא מכוון מסלול אוילר, ובמידה שכן - להחזיר אותו.

הערות:

כדי להקל על המטלה, שמות הקודקודים יתחילו מ 0 (כפי שנהוג ב Java).

קלט:

מטריצת שכנות המייצגת את הגרף.

הנחיות לתכנות:

כתבו מחלקה בשם **Euler** לחישוב מסלול אוילר בגרף. מחלקת **Euler** צריכה להכיל את הפונקציות הבאות:

1. בנאי המחלקה **public Euler (boolean[][] adj_matrix)** מקבל את מטריצת השכנות (אם יש צלע אז מופיע true).
2. פונקציה **public boolean has_euler_path()** שמחזירה האם קיים מסלול אוילר.
3. פונקציה **public String euler_path()** שמחזירה מחרוזת דוגמא למסלול אוילר (עם מפריד "->" בין זוג קודקודים). אם אין בגרף מסלול אוילר הפונקציה צריכה להחזיר מחרוזת ריקה.
4. אתם יכולים להשתמש בפונקציות \ משתנים נוספים לפי בחירתכם.

הערה מיוחדת:

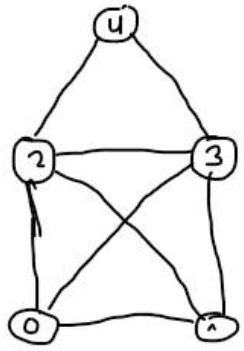
הגרף יכול להיות **כל גרף** לא מכוון.

שימו לב:

אתם לא חייבים להשתמש במבנה הנתונים שמופיע כקלט (מטריצה – מערך דו ממדי) בחישובים. אתם בהחלט יכולים להמיר אותו למבנה נתונים שיהיה לכם הרבה יותר נח (ומהיר) לעבוד איתו: להשתמש במערכים, רשימות, וקטורים, תורים, מחסניות או כל מבנה נתונים שתרצו כדי להגיע למהירות מקסימלית. תוכלו גם לבנות מחלקות חדשות משלכם, אך כולן צריכות להופיע בתוך אותו קובץ Euler.java. בנוסף, שימו לב שיש אלגוריתמים שונים בעלי סיבוכיות שונה שיכולים לעזור לכם להצליח במטלה (מומלץ מאוד לנסות כמה אלגוריתמים שתממשו ולאחר מכן לשלוח את הטוב ביותר).

ניתן להשתמש בדוגמה הבאה לבדיקת נכונות התוכנית.

Graph



Input

F	T	T	T	F
T	F	T	T	F
T	T	F	T	T
T	T	T	F	T
F	F	T	T	F

T = True

F = False

Output

has_euler_path() = True

euler_path() = 0 → 1 → 2 → 3 → 0 → 2 → 4 → 3 → 1

עבודה מהנה!