

## חלק א'

### סעיף א'

מצורף קובץ לוג גדול (logs.txt) המכיל מיליוני שורות. כל שורה מייצגת רישום לוג ומכילה קוד שגיאה. עלייך למצוא את N קודי השגיאה השכיחים ביותר.

#### משימות:

1. כתבי קטע קוד לפיצול קובץ ה-logs.txt לחלקים קטנים יותר.
  2. עבור כל חלק, ספרי את השכיחות של כל קוד שגיאה.
  3. חברי את ספירות השכיחות מכל החלקים.
  4. מצאי את N קודי השגיאה השכיחים ביותר מהספירות הממוזגות.
  5. נתחי את סיבוכיות הזמן והמקום של הפתרון שלך.
- קלט:** קובץ טקסט גדול (logs.txt) ומספר שלם N.
- פלט:** רשימה של N קודי השגיאה השכיחים ביותר וספירותיהם.

### סעיף ב'

1. מצורף קובץ נתונים גדול של סדרת זמן (time\_series.csv) המכיל שתי עמודות:  
חותמות זמן – timestamp  
ערכים מספריים – value

קלט לדוגמא:

value	Timestamp
15.3	6:10:00 2025-06-10
3.2	7:49:00 2025-06-10
12.6	7:34:00 2025-06-10
5.3	6:55:00 2025-06-10

1. כתבי קטע קוד המבצע בדיקות לפני עיבוד הנתונים כמו בדיקה של פורמט התאריך או כפילויות, חשבי על בדיקה נוספת.
  2. כתבי קטע קוד המחשב את הערך הממוצע עבור כל שעה.
- פלט רצוי:

ממוצע	זמן התחלה
10.3	6:00:00 2025-06-10
7.9	7:00:00 2025-06-10

2. בעולם בו כמות הדאטה היא עצומה – נהוג לחלק את הנתונים לחלקים קטנים ולפתור את הבעיה עבור כל חלק בנפרד, לאחר מכן מחשבים את התוצאה הסופית באמצעות איחוד כל התוצאות.

ממשי פתרון לבעיה מהסעיף הקודם באמצעות השיטה המתוארת לעיל.

כתבי קוד המחלק את קובץ ה-time\_series.csv לחלקים קטנים יותר המבוססים על זמן (למשל, חלקים יומיים).  
חשבי את הממוצעים השעתיים עבור כל קובץ בנפרד,  
ולבסוף אחדי את התוצאות וצרי באמצעותן קובץ סופי כמו בדוגמת הפלט בסעיף ב נקודה 1.

3. אם הנתונים מגיעים בזרימה (stream) במקום מקובץ, כיצד תתכנני את הפתרון כדי לעדכן את הממוצעים השעתיים בזמן אמת?

4. לתרגיל מצורף קובץ נוסף בפורמט שונה המכיל את אותו מידע, התאימי את החלקים הרלוונטיים בקוד על מנת לתמוך בפורמט החדש, וכתבי תיעוד המתאר את היתרונות באחסון המידע בפורמט הנתון.

#### קלט:

קובץ (time\_series.csv) CSV עם עמודות חותמת זמן וערך.  
קובץ (time\_series.parquet) PARQUET עם עמודות חותמת זמן וערך.  
**פלט:** קובץ CSV או פלט מובנה המכיל ממוצעים שעתיים (חותמת זמן, ערך ממוצע).

## חלק ב'

### טבלאות (כתובים בהתאמה)

Person\_Id | Personal\_Name | Family\_Name | Gender | Father\_Id |  
Mother\_Id | Spouse\_Id

ID, שם פרטי, שם משפחה, מגדר, ID של אב, ID של אם, ID של בן/בת זוג

### תרגיל 1 הקמת עץ משפחה

לבצע טבלה שיש בה כמה רשומות לכל בן אדם בהן יופיעו כל האנשים בקרבה ראשונה אליו.

הטבלה צריכה להיות בנויה כך:

Person\_Id | Relative\_Id | Connection\_Type

Connection\_Type = יכול להכיל את הערכים הבאים: אב, אם, אח, אחות, בן, בת, בן זוג ובת זוג.

### תרגיל 2 השלמת בני / בנות זוג

השלמה של בני זוג. לקחת בחשבון שיש נתונים חלקיים. כלומר, לאדם עם ID 111 מוגדר בת זוג עם ID 222. אבל, לבת זוג עם ID 222 לא מוגדר שיש בן זוג בכלל. צריך לבצע השלמה שבטבלה שיצרנו גם לבנאדם עם ID 222 יוגדר בן זוג.

**חלק ג'**

1. דרגי מ 1-6 (1- לא מעוניינת, 6- מאוד מעוניינת), האם הינך מעוניינת לעסוק בתפקידי חומרה/אלקטרוניקה (לדוג'- פיתוח מעגלים חשמליים, אינטגרציית מערכת, בדיקות חשמליות וכד')?
2. איך עובד שלט של מזגן? יש להתייחס לסעיפים הבאים (שימי לב, ניתן להציע מגוון אפשרויות למימוש אין תשובה 'נכונה'):

  - a. מהו אופן השידור בין השלט למזגן?
  - b. אילו רכיבים צריכים להיות בצד השלט ואילו בצד המזגן?
  - c. איך המזגן 'יודע' על איזה לחצן לוחצים בשלט? (במילים אחרות- איך ניתן לייצג מקשים שונים בשידור מהשלט למזגן?), יש להציע מספר אפשרויות.

## חלק ד'

### מערכת לניהול מכלות:

בעל המכולת השכונתית ביקש ממך לפתח עבורו מערכת שתסייע לו בניהול החנות. המערכת תכיל צד שרת, אליו יוכל להתחבר בעל המכולת, וצד לקוח עבור הספקים שלו.

### שלב א'

עליך לפתח את המערכת בהתאם לדרישות הבאות:

---

צד לקוח – כניסת ספק

- רישום משתמש חדש למערכת / כניסה למשתמש קיים
- צפיה בהזמנות שביצע בעל המכולת
- אישור הזמנה – הזמנה שאושרה תעבור לסטטוס "בתהליך".

בעת הרישום כל ספק ימלא את הפרטים הבאים:

- שם חברה
- מספר טלפון
- שם נציג
- רשימת הסחורות שהספק מציע (במבנה נתונים לבחירתך)

---

צד שרת – כניסת בעל המכולת

- הזמנת סחורה מספק.
- צפייה בסטטוס הזמנות קיימות
- אישור קבלת הזמנה – תעבור לסטטוס "הושלמה" ותעדכן את הספק שההזמנה התקבלה.
- מאגר של כל ההזמנות (כולל כאלו שהושלמו)

רשומה של סחורה צריכה להכיל את הנתונים הבאים:

1. שם המוצר
2. מחיר לפריט
3. כמות מינימלית לרכישה

עצבי את פורמט ביצוע הזמנה באופן הנוח ביותר.

---

דגשים

1. ניתן לבחור כל סוג מאגר מידע איתו את מרגישה בנוח לעבוד. לדוגמא: MongoDB, SQL.
2. את צד השרת ניתן לממש בכל טכנולוגיה איתה את מרגישה בנוח לעבוד. לדוגמא: Node.js, python, asp.net, java, Csharp
3. יש לממש API אליהם יוכל לפנות צד הלקוח עם route של **שליפה והכנסה** **בלבד** (אין צורך לממש עריכה ומחיקת רשומות).

### **סעיף בונוס - ניהול סחורות והזמנה אוטומטית**

בעל המכולת התלהב מהמערכת, ורוצה לנהל בעזרתה את כל הסחורות בחנות. המערכת תכיל מאגר של כלל הסחורות במכולת, ותתחבר לקופה דרך API. לכל סחורה בעל המכולת יגדיר מהי הכמות המינימלית שלה שהוא מעוניין שתהיה בחנות בכל זמן נתון. ברגע שכמות הסחורה תרד מתחת לערך הזה, המערכת תבצע הזמנה אוטומטית מהספק שמציע את המחיר הטוב ביותר עבור אותה סחורה. אם אין אף ספק שמספק את הסחורה המדוברת, היא תתריע על כך.

### **API של הקופה**

בכל פעם שתתבצע קנייה השרת יקבל קובץ JSON הבנוי באופן הבא:

```
{
  "merchandise_1": quantity,
  "merchandise_2": quantity,
  ...
}
```

לדוגמה, קנייה של 2 קרטוני חלב וכיכר לחם תראה כך:

```
{
  "bread": 1,
  "milk_carton": 2
}
```