תאריך: **03.10.2000**

שם הקורס: **שפות תוכנה מתקדמות 1**

משך הבוחן: **שלוש שעות**

חומר עזר: **כל חומר כתוב מצולם או מודפס (כולל ספרים)**

- **<u>משקל כל שאלה 50 נק' + 10 נק' בונוס</u>**

- בכל תוכנית חייבים להיות הסברים, ושמות המשתנים חייבים להיות בעלי משמעות.
- אין להוסיף פונקציות אופרטורים או משתנים פרט לאלו שנדרשים בשאלות באופן מפורש, **אלא אם נאמר אחרת!!!**

1. בסוף השאלה נתון **class BoolVector** ובו יש שני שדות נתונים ב-**private** ומספר פונקציות. ה-**class** מקבל אורך של וקטור של ערכים בוליאניים, ומצביע לוקטור.

ה-**default constructor** מקבל את אורך הוקטור ואת המצביע לוקטור ממנו הוא מקבל את הנתונים.

הפונקציה **SetVector** מעתיקה אוביקט אחד לתוך השני.

הפונקציה **GetVector** מעבירה את נתוני השדה **m_pVector** לתוך וקטור.

הפונקציה **Write** כותבת את כל האובייקט לתוך קובץ. **הכתיבה מתבצעת בסוף הקובץ**.

פונקצית עזר **EmptyVector** מציבה את ה-**class** במצב כמו של **default constructor** ללא פרמטרים.

1.1. הגדירו את האופרטורים "+", "+=", ו-"!".

"+" – מבצע פעולת **AND** בין כל שני אלמנטים בוקטור של ה-**class**-ים אם הם בעלי גודל זהה אחרת מחזיר **class** ריק.

"+=" – כמו "+" רק התוצאה נשארת בתוך ה-**class**.

"!" – לוקח כל אלמנט בוקטור ומבצע עליו פעולת **NOT** התוצאה תישמר באובייקט אחר.

1.2. שתי פונקציות **friend** כותבות את הוקטור אחת למסך והאחרת לקובץ, כך שכל האלמטים בוקטור מופרדים ב-**TAB** (**\t**), ובסוף יורדות שתי שורות. כתבו את הפונקציות.

1.3. כתבו את ה-**default constructor**, **SetVector**, **GetVector**, **Write**, ו-**EmptyVector.**

1.4. הגדירו אלו פונקציות ואופרטורים יכולים להיות **const**. **נמקו!!!**

- **אם יש צורך בפונקציות עזר יש לנמק מדוע ולרשום אותן במקום המתאים.**
- **אם יש צורך ב-constructor-ים נוספים יש לנמק למה, ולכתוב אותם.**
- **אין להוסיף משתנים נוספים ל-class.**

```
class BoolVector
{
    friend ??? operator << (???)                                                    ;
    friend ??? operator << (???)                                                    ;
  public:
    BoolVector ( const short sXSize = 0 , const bool * pBoolVector = 0 )            ;
    BoolVector & SetVector (const BoolVector & refBoolVector )                      ;
    bool * GetVector ( )                                                            ;
    BoolVector & Write ( const char * strFileName )                                 ;
    ~BoolVector ( )                                                                 ;
```

OPERATORS FOR OVERLOADING

```
  private:
    short           m_sXSize                                                        ;
    bool        *   m_pBoolVector                                                   ;
    EmptyVector ( )                                                                 ;
}
```

2. נגדיר נקודה רב-מימדית – זהו class עם מספר משתנים כמספר המימדים, כל משתנה מטיפוס float.

המרחק בין שתי נקודות יוגדר באמצעות:

$$d(\mathbf{X},\mathbf{Y})=\left(\sum_{n=1}^{N}(x_n-y_n)^2\right)^{1/2}$$

אליפסה רב-מימדית מוגדרת באמצעות שתי נקודות רב-מימדיות ($\mathbf{X_1}$ ו-$\mathbf{X_2}$) ורדיוס $radius$ מטיפוס float. התנאי לגבי הרדיוס

$$radius \geq d(\mathbf{X_1},\mathbf{X_2})$$

2.1. עליכם לכתוב שני class-ים לנקודה חד-מימדית, דו-מימדית (Point1D, ו-Point2D). לכל אחד השדות המתאימים וה-constructor המתאים.

2.2. עליכם לכתוב שני class-ים של אליפסות (Elips1D, ו-Elips2D). כל class עם השדות המתאימים וה-constructor המתאים – אשר גם יבדוק האם הרדיוס נכון. אם לא, הרדיוס יהיה המרחק בין שתי הנקודות כפול 2.

2.3. כיתבו את ה -operator (אופרטור -) עבור class-ים של Point. אופרטור זה מקבל נקודה ומחזיר את המרחק ממנה במשתנה מטיפוס float. כלומר ה -operator מיישם את הנוסחה

$$d(\mathbf{X},\mathbf{Y})=\left(\sum_{n=1}^{N}(x_n-y_n)^2\right)^{1/2}$$

.

2.4. לכל class מסעיף 2.2 כיתבו member function, המחשבת את המרחק בין המרכזים של האליפסה ומחזירה את הערך.

Float CentersDistance ( ) ;

וכן **member function** בשם **GetRadius** אשר לא מקבלת פרמטרים ומחזירה את הערך של המשתנה *radius*.

<u>שימו לב</u>: הפונקציות לכל ה-**class**-ים מטיפוס **Elips** תהינה בשם **CentersDistance** ובשם **GetRadius.**

2.5. כיתבו פונקציה חיצונית אשר בודקת האם הרדיוס גדול מפעמיים המרחק מהמרכזים ואשר מחזירה ערך בוליאני.

הפונקציה תהיה מהצורה bool RadiusSize(E??? &) ;

התוצאה תהיה true או false. פונקציה זו יכולה לקבל את כל ה-class-ים ממשפחת Elips.

- הערה: אסור למשתנים להיות ב-public.
- אם לא הצלחתם סעיף כלשהו – הניחו שהוא פתור והמשיכו הלאה.

בהצלחה