

**סדנא ב- C++ – 150018****תרגיל בית מספר 2****מחלקות עם שטחים דינאמיים****שים/י לב:**

- א. הקפד/י על קריאות התכנית ועל עימוד (Indentation).
- ב. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- ג. בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התכנית.
- ד. יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים (המופיע באתר הקורס) וביניהם:  
השתמש/י בשמות משמעותיים עבור המשתנים.  
יש לתעד את התכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים, ובנוסף, **דוגמת הרצה לכל תכנית בסוף הקובץ!**  
הגשה יחידנית - אין להגיש בזוגות.

**הערה חשובה:** לכל תרגיל בית מוגדר שבוע אחד בלבד להגשה, אלא אם כן קיבלת הוראה אחרת מהמרצה שלך. תיבות ההגשה הפתוחות לא מהוות היתר להגשה באיחור.

**שאלה מס' 1:**

- א. הגדרי/י מחלקה לייצוג נקודה במישור. הוסף/י למחלקה לפחות מתודה המחשבת את המרחק בין שתי נקודות. ניתן להשתמש במחלקה Point מתרגיל בית 1.
- ב. הגדרי/י מחלקה לייצוג מצולע כלשהו במישור. המחלקה תכלול את השדות הבאים:
- מצביע למערך של קדקודים (נקודות במישור – סעיף א).
  - מספר הקדקודים במצולע.
- הוסף/י למחלקה לפחות את המתודות הבאות:
- constructor
  - copy-constructor
  - destructor
  - מתודה addPoint המקבלת אינדקס וערכי קודקוד ומוסיפה קודקוד זה במערך הקודקודים
  - מתודה המחשבת את היקף המצולע (סכום המרחקים בין כל שני קודקודים סמוכים). ניתן להניח שכל הנקודות מסודרות לפי הסדר בו הן מרכיבות את המצולע.
  - מתודה בוליאנית המקבלת מצולע ובודקת האם המצולע שהתקבל והמצולע הנוכחי זהים. מצולעים יוגדרו זהים כאשר מספר הקודקודים שלהם שווה וערכי הקודקודים שלהם שווים. שימו לב, לא מחייב שהסדר של הקודקודים השמורים במערך יהיה זהה.
- ג. לדוג': המצולע: (0,0) (1,1) (2,0) (0,0) זהה למצולע: (0,0) (1,1) (2,0) (1,1)
- לצורך הבנה עמוקה של מתודות ה-ctor עליך להוסיף את ההדפסות הבאות:
- constructor: in constructor
  - copy-constructor: in copy-constructor
  - destructor: in destructor

ד. כתוב/י תכנית ראשית הקולטת נתונים על שני מצולעים ומדפיסה את היקפם, **מעוגל למספר השלם הקרוב ביותר**, באופן הבא: במידה והמצולעים זהים יש להדפיס **equal** ואת היקפם. במידה ואינם זהים יש להדפיס עבור כל אחד מהם את היקפו. התכנית תדפיס עבור כל מצולע: **enter number of sides:** ותקלוט את מספר הצלעות. לאחר מכן התכנית תדפיס: **enter the point values:** ותקלוט את שיעורי הנקודות. הקלט יהיה מהצורה:  $(x_1, y_1) (x_2, y_2) \dots (x_N, y_N)$  כאשר: N הוא מספר הקודקודים במצולע, ו:  $x_i, y_i$  הם הקואורדינטות של הנקודות במצולע. בכל מקרה של קלט לא תקין התכנית תדפיס **ERROR**.

דוגמאות להרצת התכנית:

משולשים ישרי זווית שצלעותיהם 3-4-5:

ריבוע שצלעו 2 ומשולש:

```
enter number of sides:
3
in constructor
enter the point values:
(10,10) (10,14) (13,10)
enter number of sides:
3
in constructor
enter the point values:
(13,10) (10,10) (10,14)
in copy-constructor
in destructor
equal. perimeter: 12
in destructor
in destructor
```

```
enter number of sides:
4
in constructor
enter the point values:
(0,0) (0,2) (2,2) (2,0)
enter number of sides:
3
in constructor
enter the point values:
(1,1) (2,0) (3,1)
in copy-constructor
in destructor
perimeter: 8
perimeter: 5
in destructor
in destructor
```

תזכורת: בכדי להשתמש בפונקציות מתמטיות יש להכליל את הספריה `cmath`.

שאלה מס' 2:

הגדר/י מחלקה בשם Vector למימוש מערך של מספרים שלמים באורך כלשהו.

המחלקה תכלול את השדות הבאים:

- data - מצביע למערך של שלמים
- capacity - אורך הוקטור (מספר האיברים המקסימלי העכשווי במערך)
- size - מספר הערכים הנמצאים בפועל בוקטור

הוסף לפחות את המתודות הבאות:

- constructor - מקבל כפרמטר את גודל הוקטור המבוקש ויוצר אותו. אם לא התקבל ערך, הקונסטרוקטור יציב ערך ברירת מחדל 2, כאשר size מגיע לגודל ה-capacity יש להגדיל את המערך פי שניים מגודלו בזמן ההוספה (ולבצע העתקה ושחרור בהתאם).
- copy constructor.
- destructor.
- מתודה בשם getCapacity() המחזירה את מספר האיברים המקסימלי בוקטור בעת הקריאה למתודה.
- מתודה בשם getSize() המחזירה את מספר האיברים הקיימים בפועל בוקטור.
- מתודה בשם print(). המתודה תדפיס את נתוני הוקטור: גודל מקסימלי, גודל בפועל, והערכים בוקטור.
- לדוגמא וקטור באורך 4 שבו שני תאים מלאים כרגע (5,19) יודפס כך:  
**capacity: 4 size: 2 values: 5 19**
- מתודה בשם assign (מתפקדת כאופרטור =) המקבלת כפרמטר וקטור נוסף ומשנה את הוקטור הנוכחי להיות זהה לוקטור שהתקבל.
- מתודה בשם isEqual (מתפקדת כאופרטור ==) המקבלת כפרמטר וקטור נוסף ומחזירה true כאשר שני הוקטורים זהים הן בגודלם הנוכחי המלא (size) והן בתכנם. אחרת תחזיר false.
- מתודה בשם at המקבלת אינדקס (מתפקדת כאופרטור []) **להצבה והחזרה** של ערך בתא מבוקש בוקטור. אם התא המבוקש אינו בטווח התאים המלאים במערך יש להדפיס ERROR ולהחזיר כפרמטר את האיבר הראשון. כותרת המתודה: `int& at(int index);`
- מתודה בשם strcatcat המקבלת כפרמטר וקטור נוסף ומבצעת מכפלה סקלרית בין שני וקטורים (יש להכפיל ערך כל תא בוקטור הראשון עם הערך בתא המקביל בוקטור השני ולסכום את התוצאות). אם בשני הוקטורים אין מספר זהה של ערכים יש להדפיס **ERROR** ולהחזיר -1.
- מתודה בשם strnewcat לשרשור שני וקטורים. המתודה מקבלת וקטור נוסף כפרמטר ומייצרת (ומחזירה) וקטור חדש שתוכנו שרשור של 2 הוקטורים, כאשר הנוכחי ראשון והאחד שהתקבל שני.
- מתודה בשם clear() שתפקידה לרוקן את וקטור (ריקון ערכים).
- מתודה בשם delLast() המוחקת את האיבר האחרון בוקטור. במידה והוקטור ריק (חמיקה) יש להדפיס **ERROR**.
- מתודה בשם insert(int val) המציבה את הערך val במקום הבא הפנוי במערך. במידה והמערך מלא יש להגדיל את המערך (באופן דינאמי) פי שניים מגודלו הנוכחי, יש להעתיק את הערכים מהמערך המקורי למערך המוקצה החדש ולשחרר את המערך המקורי מהזיכרון.

יש להשתמש בתכנית הראשית הבאה בכדי לבחון את נכונות המתודות שכתבת:

```
#include "Vector.h"
#include <iostream>
using namespace std;
enum options
{
    stop, assignment, isEqual, mult, add, clear, delLast, at, insert
};
int main()
{
    Vector v1(10), v2(10), v3;
    for (int i = 1; i <= 4; i++)
    {
        v1.insert(i);
        v2.insert(i + 4);
    }
    int choice, val, index;
    cout << "enter your choice 0-8:\n";
    cin >> choice;
    while (choice)
    {
        switch (choice)
        {
            case assignment: v3.assign(v1);
                            break;
            case isEqual: if (v1.isEqual(v2)) cout << "v1==v2\n"; else cout
<< "v1!=v2\n";
                            break;
            case mult:      cout << "v1*v2=" << v1.strcatcat(v2) << endl;
                            break;
            case add:       v3.assign(v1.strnewcat(v2));
                            break;
            case clear:     v1.clear();
                            break;
            case delLast:   v2.delLast();
                            break;
            case at:        cout << "enter index:" << endl;cin >> index;
                            cout << "enter value:" << endl; cin >> val;v3.at(index) = val;
                            break;
            case insert:    cout << "enter value:" << endl;cin >>
val;v3.insert(val);
                            break;
            default: cout << "ERROR";
        }
        v1.print();      v2.print();      v3.print();
        cout << "enter your choice 0-8:\n";
        cin >> choice;
    }
    return 0;
}
```

דוגמאות להרצת התכנית:

enter your choice 0-8:

3

v1\*v2=70

capacity: 10 size: 4 values: 1 2 3 4

capacity: 10 size: 4 values: 5 6 7 8

capacity: 2 size: 0 values:

enter your choice 0-8:

0

enter your choice 0-8:

4

capacity: 10 size: 4 values: 1 2 3 4

capacity: 10 size: 4 values: 5 6 7 8

capacity: 20 size: 8 values: 1 2 3 4 5 6 7 8

enter your choice 0-8:

0