

סדנא ב- C++ – 150018

תרגיל בית מספר 5

מחלקה מוכלת – רשימה לינארית

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:

- השלם/י את הגדרת המחלקה List שהוצגה בהרצאה, כך שתאפשר לטפל ברשימה המכילה איברים ממיונים בסדר יורד (כל איבר ברשימה קטן ממש מהאיבר הקודם לו). הוסף/י את המתודות הבאות:
- א. operator= - העתקה עמוקה של רשימה אחת לתוך השניה בסדר זהה.
 - ב. מתודת פלט - << operator להדפסת כל אברי הרשימה.
 - ג. מתודת קלט - >> operator לקליטת אברי המערך בצורה ממיונת, כלומר, הקלט יסתיים כאשר יתקבל ערך שאינו קטן (ממש) מהערך הקודם לו.
 - ד. insert (int key) מתודה המקבלת מספר שלם key ומכניסה אותו למקום המתאים ברשימה הממיונת. בשלב זה הרשימה יכולה להיות ממיונת בסדר **לא עולה** (ייתכנו איברים כפולים).
 - ה. מתודת מחיקה - remove (int key). המתודה מקבלת מספר שלם key. במידה והערך key קיים ברשימה, המתודה מוחקת אותו מהרשימה תוך שמירה על המיון של הרשימה. במידה והערך לא קיים, תשלח הודעת חריגה "value not found". במידה והערך קיים יותר מפעם אחת, יש למחוק את המופע הראשון שנתקלים בו.

נתונה התכנית הראשית הבאה, הבוחנת את נכונות המחלקה:

```
#include <iostream>
#include "List.h"
using namespace std;
```

```
int main()
{
    List lst;
    int choice, val;
    cout<<"enter the list values\n";
    cin>>lst;
    cout<<"choose 0-2\n";
    cin>>choice;
    while (choice)
    {
        switch (choice)
        {
            case 1:cout<<"enter a value to insert\n";
                    cin>>val;
                    lst.insert(val);
                    break;
            case 2:cout<<"enter a value to remove\n";
                    cin>>val;
                    try{
                        lst.remove(val);
                    }
                    catch(char * msg)
                    {
                        cout<<msg<<endl;
                    }
                    break;
            default:cout<<"ERROR\n";
        }
        cout<<lst<<endl;
        cout<<"choose 0-2\n";
        cin>>choice;
    }
    return 0;
}
```

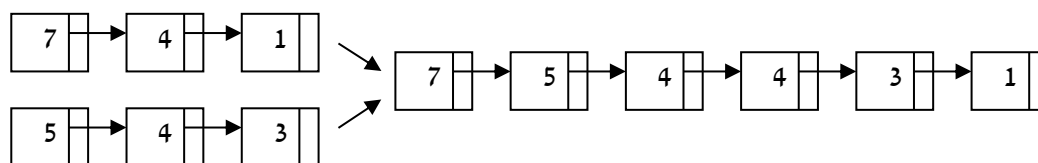
דוגמה להרצת התכנית הרצה :

```
enter the list values
3 2 1 1
choose 0 - 2
2
enter a value to remove
2
3 1
choose 0 - 2
1
enter a value to insert
4
4 3 1
choose 0 - 2
1
enter a value to insert
2
4 3 2 1
choose 0 - 2
0
```

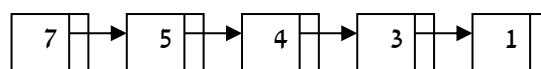
שאלה מס' 2:

השתמש/י במחלקה List המטפלת ברשימה ממוינת (המוגדרת לעיל), וכתוב/י את שלושת הפונקציות **הגלובליות** הבאות (שים/י לב – אין ליצור פונקציות חברות של המחלקה, אלא ליצור את הפונקציות ב-main ולהיעזר במתודות public בלבד):

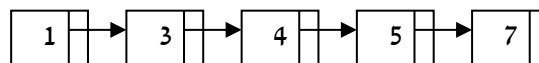
א. פונקציה merge למיזוג שתי רשימות. הפונקציה מקבלת שתי רשימות ממוינות lst1 ו- lst2 מטיפוס List, ומחזירה רשימה חדשה הבנויה ממיזוג שתי הרשימות בסדר **לא עולה**. (שים/י לב – ברשימה החדשה ייתכנו איברים כפולים)



ב. פונקציה makeSet להפיכת רשימה לקבוצה. הפונקציה מקבלת רשימה ממוינת בסדר **לא עולה**. הפונקציה משנה את הרשימה, כך שכל איבר יופיע בה פעם אחת בלבד (לאחר ביצוע הפונקציה, לא יהיו ברשימה איברים כפולים – הרשימה תהיה ממוינת בסדר **יורד**).



ג. פונקציה reverse ההופכת את הרשימה, כלומר האיבר האחרון יצביע לאיבר שלפניו וכו' עד כאשר האיבר הראשון יצביע ל-null. (מה שיגרום לרשימה ממוינת בסדר **לא יורד**)



נתונה התכנית הראשית הבאה, הבוחנת את נכונות הפונקציות:

```

#include <iostream>
#include "List.h"
using namespace std;
// הגדרת ומימוש הפונקציות

int main()
{
    List lst1, lst2, mergedList;

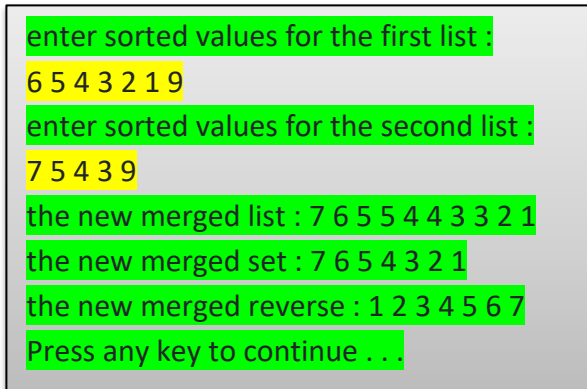
    cout<<"enter sorted values for the first list:"<< endl;
    cin>>lst1;
    cout<<"enter sorted values for the second list:"<< endl;
    cin>>lst2;

    mergedList = merge(lst1,lst2);
    cout <<"the new merged list: " << mergedList <<endl;

    makeSet(mergedList);
    cout<<"the new merged set: " << mergedList << endl;
}
  
```

```
reverse(mergedList);  
cout<<"the new merged reverse: " << mergedList << endl;  
  
return 0;  
}
```

דוגמה להרצת התכנית:



```
enter sorted values for the first list :  
6 5 4 3 2 1 9  
enter sorted values for the second list :  
7 5 4 3 9  
the new merged list : 7 6 5 5 4 4 3 3 2 1  
the new merged set : 7 6 5 4 3 2 1  
the new merged reverse : 1 2 3 4 5 6 7  
Press any key to continue . . .
```

בהצלחה!