

תכנות מונחה עצמים

עבודת הגשה מספר 3

מועד הגשה: עד 7.5.2017 בשעה 23:50

הוראות הגשה:

1. אנא קראו בעיון את כל תיאור העבודה בטרם תתחילו לכתוב קוד.
2. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
3. יש לוודא שהעבודה מתקמפלת על מחשבי המכללה בטרם ההגשה. עבודה שלא מתקמפלת תקבל ציון 0.
4. בתחילת הקובץ יש להוסיף את התיעוד הבא:

/* Assignment: 3

Campus: Ashdod / Beer Sheva (תבחרו את המתאים)

Author: Israel Israeli, ID: 01234567 */

כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.

5. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
6. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
7. אין להשתמש בתרגיל זה בחומר שטרם נלמד בהרצאות. אין להשתמש בהורשה.
8. למחלקות המפורטות בעבודה ניתן להוסיף מתודות נוספות אך אסור להוסיף שדות.
9. יש להכניס את החלק התיאורטי בקובץ וורד נפרד. יש להכניס את כל הקבצים של המשימה הראשונה מהחלק המעשי לתיקייה בשם part1, כל הקבצים של משימה השנייה לתיקייה בשם part2, כל הקבצים מסעיף הבנוס (לא חובה) לתיקייה בשם part3 ואז לכווץ יחד. נדרש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים של כל השאלות. לקובץ המכווץ יהיה שם המהווה את מספר ת.ז. של המגיש.
10. שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראית לתרגיל, סבטלנה רוסיין, במייל: sceassign2016@gmail.com.
11. הארכות יינתנו אך ורק במקרים חריגים (מילואים, אבל על קרובים ומחלה חריפה!) ובצורף אישורים מתאימים! כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה האחראית על התרגיל לפחות יומיים לפני חלוף הדד-ליין!

הערות:

1. יש להקפיד על תכנות נכון:
 - a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: const, define או enum, בהתאם לצורך.
 - b. להוסיף מילה const בכל מקום בקוד שהדבר אפשרי.
 - c. יש לרשום הערות בשפוע! ובאנגלית בלבד (לכל מחלקה מה התפקיד שלה, התוכנית מה היא מבצעת, כל פעולה לא טריוויאלית להסביר – הערה, כל מתודה מה היא עושה).

- d. יש להקפיד על כימוס נכון – כל השדות ומתודות השירות ב-private, והממשק ב-public! חלוקה לקבצים!
- e. יש להקפיד על הזחות! כיתוב נכון וקריא! ושמות משמעותיים!
- f. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על reuse בקוד.
- g. לפני בקשת קלט (cin) יש להדפיס למשתמש הוראה (cout) איזה קלט מבוקש.
- h. יש להקפיד על מוסכמות התכנות הנכון (שמות כמו שצריך וכו').
- i. יש להקפיד על כל כללי התכנות הנכון כפי שנלמדו בכיתה.
- j. שימו לב שגתינת הרשאות friend פוגעת בכימוס! ועל כן יש למעט בשימוש בה ולעשות זאת רק במצבים שבהם זה נדרש - כפי שלמדנו בכיתה.

(1 חלק א' – תאורטי (מענה בקובץ טקסט – וורד): 7 נקודות)

- (1) מה היא המוטיבציה להעמסת אופרטור כפונקציית friend? יש לתת דוגמה.
- (2) האם ניתן ליצור במחלקה אופרטור מסוג static? יש לתת הסבר קצר.
- (3) מה הם היתרונות של האופרטורים new ו-delete לעומת פונקציות malloc ו-free?
- (4) איך ניתן להמיר את הטיפוס הסטנדרטי (int, float, ...) לסוג האובייקט (user-defined type)? ואיך נבצע את הפעולה ההופכית? יש לתת דוגמאות קצרות.
- (5) האם הטענה הבאה היא טענה נכונה: מתודה-אופרטור להבדיל ממתודה רגילה יכולה להחזיר reference לאובייקט לוקלי שלה. יש לתת נימוק לתשובה.
- (6) יש להשלים את הקוד ולממש בנאי העתקה ע"י שימוש באופרטור השמה. במידה ונדרש יש לעדכן חלקים נוספים באופרטור ההשמה.

```
class Worker{
    int id;
    char* name;
public:
    .....
    Worker (const Worker & );
};
```

```

Worker & operator=(const Worker & );

};

Worker:: Worker (const Worker & other ){

    //implementation by assignment operator

}

Worker & Worker::operator=(const Worker & other ){

    if(this != & other){

        id=other.id;

        delete[] name;

        name=new char[strlen(other.name)+1];

        strcpy(name,other.name);

    }

    return *this;

}

```

חלק ב' – מעשי (ההגשה היא של קבצי ה- CPP ו- H בלבד) – 93 נקודות:

1. משימה ראשונה – משחק זיכרון (53 נקודות) :

– שמות הקבצים שיוגשו הם

Card.h
Card.cpp
Pile.h
Pile.cpp
Game.h
Game.cpp
Driver.cpp

תיאור העבודה:

משחק הזיכרון הינו משחק חשיבה לילדים אשר מטרתו לתרגל ולשפר זיכרון אנושי על-ידי התאמת זוגות קלפים בחבילה.

במשחק זה, המשתתפים פורסים חבילת קלפים, המורכבת מזוגות זהים של קלפים הפוכים. כל שחקן בתורו מרים שני קלפים במטרה למצוא התאמה. אם שני הקלפים זהים, השחקן ייקח אליו את הקלפים והם ייצברו לזכותו, אחרת יחזיר את הקלפים למקומם כאשר פניהם כלפי מטה. בתחילת המשחק, סביר כי יתקשו המשתתפים למצוא שני קלפים זהים, אך ככל שמספר הניסיונות עולה, כך המשתתפים זוכרים את מאפייני הקלף ומיקומו.

במטלה זו, יש לבנות משחק זיכרון לשחקן יחיד.

תהיינה המחלקות הבאות:

1. מחלקת קלף – לכל קלף יש תו ומצב - פתוח או סגור.

המחלקה תכיל לפחות את הפונקציות הבאות:

בנאי.

אופרטור ! להפיכת מצב הקלף (אם היה פתוח יהיה סגור וההיפך).

אופרטור == להשוואה (שבדוק האם 2 קלפים מכילים אותיות זהות).

אופרטור << (OUT) להדפסת קלף בצורה הבאה: לדוגמא כאשר הקלף הינו A, יופיע [A] כאשר הקלף פתוח ו- [?] כאשר הקלף סגור.

2. מחלקת ערימה (חבילה) – ערימה היא מערך דינמי של קלפי זיכרון. (חובה שיהיה מצביע ולא מערך בגודל קבוע).

הערימה תכיל תמיד קלפים במספר זוגי (במידה ונמסר מספר אי-זוגי יש להכניס את המספר קטן באחד – ז"א האם נמסר מספר 25 – החבילה תכיל 24 קלפים) מספר מקסימלי של קלפים - 52.

המחלקה תכיל לפחות את הפונקציות הבאות:

בנאי – אמור:

1. לבחון האם גודל הערימה חוקי (עד 52 קלפים מקסימום), במידה ומוזן מספר לא חוקי יש ליצור מערך בגודל 20. כמו-כן, על המערך להיות בגודל זוגי, אחרת יש להקטינו באחד.

2. ליצור מערך דינמי מתאים, למלא ולערבב אותו.

אופרטור [] המקבל אינדקס ומחזיר את האיבר שנמצא במיקום האינדקס בקופה. אם ערך האינדקס אינו חוקי עבור הקופה, יש להוציא הודעה מתאימה ולצאת מהמשחק.

אופרטור << (OUT) להדפסת כל הקופה לפי מצבה. יש לפרוס את הקלפים כך שבכל שורה לא יהיו יותר מ- 5 קלפים.

פונקציה למילוי הקופה – הפונקציה תמלא את הקופה כך שלכל קלף יוצב אות מה-ABC. ותמיד יהיו 2 קלפים זהים.

פונקציה לערבוב הערימה – ערבוב אקראי של מיקומי הקלפים (Index), הצגה של הקלפים למשך 30 שניות (הקוד לביצוע ההשהיה מופיע בנספח), לאחר ההצגה יש להפוך את כל החבילה.

הורס.

3. מחלקת משחק – תכיל ערימה אחת ומשתנה עבור מספר הזוגות שהשחקן גילה.

בנאי.

למחלקה יש רק פונקציה ציבורית אחת **Run** – פונקציה זו תנהל את כל המשחק.

כל שאר הפונקציות של מחלקה זו יהיו פרטיות.

לכל המחלקות ניתן להוסיף פונקציות ככל שיידרש.

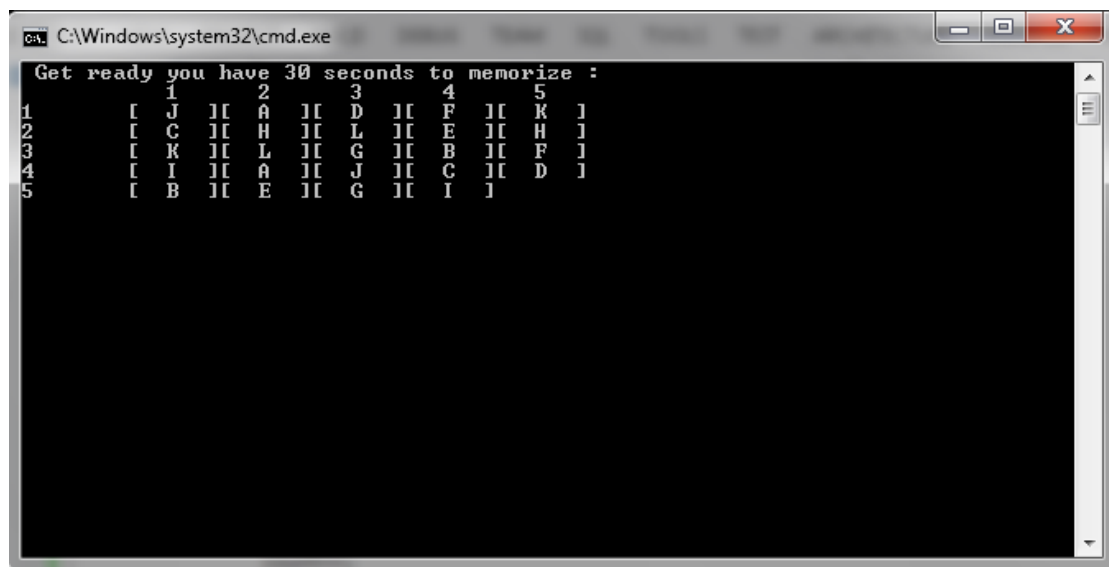
התוכנית הראשית נראית כך:

```
int main()
{
    int no;
    cout<<"Enter number of cards you want to play with"<<endl;
    cin>>no;
    Game m(no);
    m.Run();
    return 0;
}
```

מהלך המשחק:

-שימו לב כי מסכים אלו מתייחסים לערימה של 24 קלפים.

בתחילת המשחק התוכנית מציגה את כל הקלפים גלויים במשך 30 שניות, לאחר-מכן הקלפים מתהפכים והמשחק מתחיל.



לאחר פריסת הקלפים יש לאפשר לשחקן לבחור קלף ראשון לגילוי. לאחר הדפסת הלוח כאשר הקלף בו הוא בחר גלוי יש לאפשר לשחקן לבחור קלף שני לגילוי.

```

C:\Windows\system32\cmd.exe

1      2      3      4      5
1  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the first card by entering the row and col numbers
row: 1
col: 1
1      2      3      4      5
1  [ J ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the second card by entering the row and col numbers
row: _

```

ושבו להדפיס את הלוח עם שני הקלפים בהם הוא בחר גלויים.

אם הקלפים שונים יש להוציא הודעה על כך ולשאול אם ברצונו להמשיך את המשחק.

```

C:\Windows\system32\cmd.exe

1      2      3      4      5
1  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the first card by entering the row and col numbers
row: 1
col: 1
1      2      3      4      5
1  [ J ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the second card by entering the row and col numbers
row: 5
col: 1
1      2      3      4      5
1  [ J ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ B ] [ ? ] [ ? ] [ ? ] [ ? ]
No match. do you want to continue? 1=yes,0 - no

```

בכל שלב יש לשאול את שחקן האם הוא רוצה להמשיך במשחק.

```

C:\Windows\system32\cmd.exe

      1      2      3      4      5
1  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the first card by entering the row and col numbers
row: 1
col: 1

      1      2      3      4      5
1  [ J ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the second card by entering the row and col numbers
row: 5
col: 1

      1      2      3      4      5
1  [ J ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ B ] [ ? ] [ ? ] [ ? ] [ ? ]
No match. do you want to continue? 1-yes,0 - no
1

```

אם התשובה חיובית יש להדפיס שוב את הלוח עם שני הקלפים שהפכנו, בחזרה במצב סגור.

```

C:\Windows\system32\cmd.exe

      1      2      3      4      5
1  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
2  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5  [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the first card by entering the row and col numbers
row:

```

אם שני הקלפים זהים, יש להדפיס את הלוח שוב כאשר שני הקלפים גלויים.

```

C:\Windows\system32\cmd.exe

1      1      2      3      4      5
2      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the first card by entering the row and col numbers
row: 5
col: 1
1      1      2      3      4      5
2      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the second card by entering the row and col numbers
row: 3
col: 4
1      1      2      3      4      5
2      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Match!!! do you want to continue? 1=yes,0 - no
_

```

בשלב הבא הקלפים שגילנו יישארו גלויים עד סוף המשחק.

```

C:\Windows\system32\cmd.exe

1      1      2      3      4      5
2      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
3      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
4      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
5      [ ? ] [ ? ] [ ? ] [ ? ] [ ? ] [ ? ]
Choose the first card by entering the row and col numbers
row: _

```

הערה: יש לשים לב שמיקום הקלף הנמצא בשורה 1 ועמודה 1 בערימה בפועל הוא 0 ומיקום הקלף הנמצא בשורה 4 ועמודה 3 בערימה בפועל הוא 17 (בדוגמא זו).

המשחק נגמר בניצחון כאשר כל הקלפים נפתחים (כלומר השחקן מצא את כל הזוגות).

נספחים למשימה הראשונה

1. קוד לשעון שהייה בתחילת המשחק עבור sec שניות:

```
#include <ctime>
...
int sec;
clock_t start_time = clock();
clock_t end_time = sec * 1000 + start_time;
while(clock() != end_time)
    ;
```

2. ניקוי מסך:

```
#include <stdlib.h>
...
system("cls");
```

2. משימה שנייה – מניפולציות על קבוצות בעזרת רשימה מקושרת (40 נקודות):

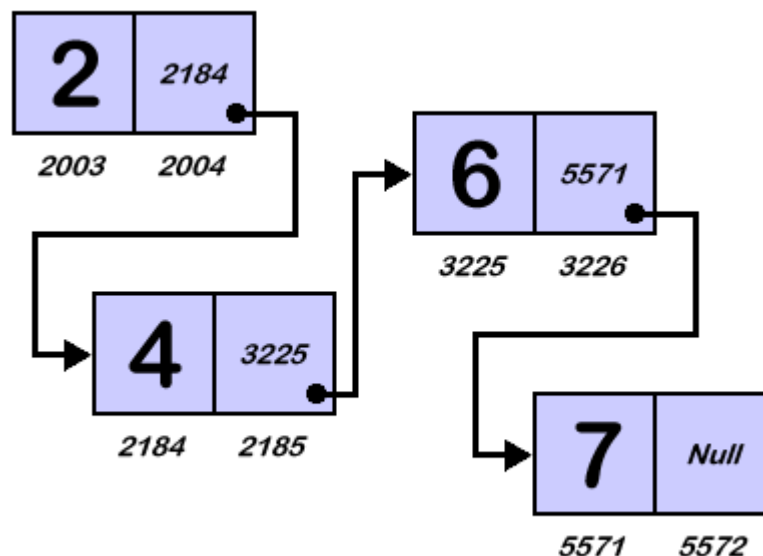
– שמות הקבצים שיוגשו הם

SetLinkedList.h
SetLinkedList.cpp
Tester.cpp (main)

הקובץ Tester.cpp מופיע בהמשך השאלה, אתם נדרשים להגיש אותו ללא שום שינוי.

תיאור העבודה:

קבוצה הינה אוסף מוגדר היטב של איברים ייחודיים. נניח כי ישנה קבוצת מספרים שלמים מסוימת $A=(2,4,6,7)$ המיוצגת על-ידי סוגריים עגולים, ללא חשיבות לסדר כלשהו, (במתמטיקה קבוצה מיוצגת על-ידי סוגריים מסולסלים). את קבוצה זו נרצה לייצג בתור מבנה נתונים מסוג רשימה מקושרת. רשימה מקושרת, הינה מבנה נתונים אשר מורכב ממספר צמתים, כל צומת מכילה מידע ומצביע לצומת הבאה (בצומת האחרונה נשמר ערך NULL).



יש ליצור מחלקה בשם **SetLinkedList** , המייצגת קבוצת מספרים שלמים בעזרת רשימה מקושרת.

יש ליצור עבור המחלקה :

- בנאי ברירת מחדל (היוצר קבוצה ריקה)
- בנאי מעתיק (יוצר קבוצה חדשה כעותק של קבוצה-פרמטר),
- הורס שמשחרר את הקבוצה (את כל הצמתים של הרשימה)
- אופרטור השמה = המעדכן את האובייקט מסוג ה"ל הנוכחי (LValue) עי" אובייקט הפרמטר (RValue).

יש ליצור את האופרטורים הבאים עבור המחלקה לטיפול בקבוצות:

1. אופרטור $+=$, אשר מקבל מספר שלם ומוסיף אותו לתחילת הקבוצה רק במידה והמספר אינו קיים בקבוצה, אחרת לא תתבצע אף פעולה נוספת.
לדוג' בהינתן הפעלת אופרטור זה על קבוצה A עם המספר 8, הרשימה החדשה תהינה :
(8,2,4,6,7).
2. אופרטור $-=$, אשר מקבל מספר שלם, במידה ומספר זה מופיע בקבוצה, אזי הוא יוציא אותו מקבוצה זו.
3. אופרטור $+$, אופרטור הפועל על קבוצה ומקבל כפרמטר קבוצה נוספת, תוצאת האופרטור הינה קבוצה חדשה המתקבלת כאיחוד שתי הקבוצות (כלל האיברים הייחודיים משניהן).
לדוג', איחוד של (18,3) ו-(3,1) היא הקבוצה (3,1,18).
4. אופרטור $-$, אופרטור הפועל על קבוצה ומקבל כפרמטר קבוצה נוספת, תוצאת האופרטור הינה קבוצה חדשה המורכבת מכלל האיברים שנמצאים בקבוצת הבסיס ואינם כלולים בקבוצת הפרמטר.
לדוג', קבוצת הבסיס הינה (10,7,8,1), וקבוצת הפרמטר (8,7,3) כך שהאופרטור יחזיר תהינה הקבוצה (10,1).
- 5.
6. אופרטור -- **(Prefix)** המוחק את האיבר הראשון בקבוצה .
7. אופרטור -- **(Postfix)** המוחק את האיבר האחרון בקבוצה.
- 8.
9. אופרטור $==$ הבודק האם 2 קבוצות זהות (ז"א הם מכילים את אותם האיברים , אין חשיבות לסדר ההופעה).
10. אופרטור $!=$ הבודק האם 2 קבוצות שונות (ז"א הם מכילים לפחות איבר אחד שונה , אין חשיבות לסדר ההופעה).
11. אופרטור פלט $<<$ למסך אשר מדפיס קבוצה באופן מסודר, כפי שמופיע בדוגמא מטה.

ניתן להוסיף שיטות פרטיות כרצונכם.

אנו נגדיר בתוך מחלקת רשימה (קבוצה) מחלקה פנימית פרטית (private inner class) בשם Node (קודקוד/צומת).

שימו לב כי מחלקה פנימית היא מחלקה שההצהרה שלה נמצאת בתוך מחלקה אחרת – מחלקת האם שלה (במקרה זה מחלקת רשימה/קבוצה). ולכן כל גישה אליה היא רק דרך מחלקת האם.

למחלקת Node יהיו שתי שדות:

- מספר מהקבוצה.
- מצביע ל-Node (next).

להלן הגדרה **חלקית** של מחלקות (אסור להוסיף שדות מעבר למוגדר !) :

```
class SetLinkedList {
    class Node{
        int data;
        Node* next;
    public:
        Node(int a):data(a),next(NULL){}
        Node(const Node &);

}; // end of class Node
Node* list_head;
public:

}; // end of class SetLinkedList

SetLinkedList::Node::Node(const SetLinkedList::Node & other): data ( other.data) {
// if there's another node after other recursively call copy constructor to
create new copy of node after other
    if (other.next!=NULL )
        next=new Node( *(other.next) ) ;
    else
        next=NULL;
}
}
```

דוגמא לשימוש במחלקה SetLinkedList (driver) :

```
int main()
{
    SetLinkedList list1;
    for(int i=20;i<100;i+=30)
        list1+=i;
    cout<<" list1: "<<list1<<endl;    // list1: ( 80 , 50, 20 )
    SetLinkedList list2(list1);
    list2+=200;
    list2+=300;
    cout<<" list2: "<<list2<<endl;// list2: ( 300 , 200 , 80 , 50 , 20 )
    for(int i=25;i<60;i++)
        list1-=i;
    list1+=55;
    cout<<endl<<endl;
    cout<<" list1 : "<<list1<<endl;// list1 : ( 55 , 80 , 20 )
    cout<<" list2 : "<<list2<<endl;// list2 : ( 300 , 200 , 80 , 50 , 20 )
    SetLinkedList temp;
    temp=list1+list2;
    cout<<" Union : "<<temp;// Union : ( 55 , 80 , 20 , 300 , 200 , 50 )
    cout<<endl<<endl;
    cout<<" list1 : "<<list1<<endl;    // list1 : ( 55 , 80 , 20 )
    cout<<" list2 : "<<list2<<endl;// list2 : ( 300 , 200 , 80 , 50 , 20 )
    temp=list1-list2;
    cout<<" ----- Minus : "<<temp;// ----- Minus : ( 55 )
    cout<<endl<<endl;
    SetLinkedList list3(list2);// 300 200 80 50 20
}
```

```

    cout<<" prefix --:";
    cout<<"list3:"<<--list3<<endl;// prefix --:list3: ( 200 , 80 , 50 , 20 )
    cout<<"list3 after prefix : "<<list3<<endl;
// list3 after prefix : ( 200 , 80 , 50 , 20 )
    cout<<" postfix --:";
    cout<<"list2 : "<<list2--<<endl;
// postfix --:list2 : ( 300 , 200 , 80 , 50 , 20 )
    cout<<"list2 after postfix : "<<list2<<endl;
// list2 after postfix : ( 300 , 200 , 80 , 50 )
    cout<<endl<<endl;
    SetLinkedList list4;
    list4+=50;
    list4+=200;
    list4+=300;
    cout<<" list 4 : "<<list4<<endl;    // list 4 : ( 300 , 200 , 50 )
    if(list4==list2)
        cout<<" equals";
    else
        cout<<" not equals";           //not equals
    cout<<endl<<endl;
    list4+=80;
    if(list4==list2)
        cout<<" equals";               //equals
    else
        cout<<" not equals";
    cout<<endl<<endl;
    if(list1!=list2)
        cout<<" not equals";           //not equals
    else
        cout<<" equals";
    cout<<endl<<endl;
    return 0;
}

```

עליכם לשלב את התהליך המוצג בדרייבר ומבנה המחלקה החלקי עם הפונקציונליות הנדרשת לטובת המשימה.

סעיף בונוס (20 נקודות)

- הוסף את הפונקציות הבאות לתכנית של חלק 2:
1. כתוב פונקציה ההופכת את הקבוצה (רשימה) כך שאיברי הרשימה יקושרו בכיוון הפוך – מהסוף להתחלה.
 2. כתוב פונקציה שממיינת רשימה מקושרת בסדר יורד לפי ערך הנשמר בצומת.

להלן חלק מהדרייבר הרלוונטי :

```

cout<<endl<<list4; // ( 80 , 300 , 200 , 50 )
list4.Reverse();
cout<<endl<<list4;    // ( 50 , 200 , 300 , 80 )
SetLinkedList listTest;
listTest+=15;
listTest+=10;
listTest+=30;
listTest+=25;
cout<<endl<<listTest; // ( 25 , 30 , 10 , 15 )
listTest.Sort();
cout<<endl<<listTest; // ( 30 , 25 , 15 , 10 )

```

בהצלחה ☺