

מבחן סוף סמסטר

מועד ב'

בבחינה זו 4 שאלות ו- 8 עמודים.
משך הבחינה: שלוש שעות

נא לרשום את השם ומספר תעודת הזהות במקום המיועד במחברת הבחינה.

עליכם לענות על כל השאלות אך ורק במחברת הבחינה.

יש להגיש רק את מחברת הבחינה.

מומלץ לקרוא כל שאלה בעיון רב לפני שניגשים לפתור אותה.

כל חומר עזר מותר.



בהצלחה!!!

שאלה מס' 1 (25 נק')

אנו רוצים לממש מטריצה 2×2 ע"י שימוש בשפת C ובמתודולוגית טיפוסית נתונים מופשטים (ADT). מטריצה $X2$

$$2 \text{ ניתנת להצגה כך: } M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

D, C, B, A הם מספרים ממשיים (מסוג double).

על מטריצה 2×2 מוגדרות הפעולות הבאות:

* בכל הפעולות (פרט לפעולת האתחול) אין לשנות את מטריצות הקלט, ובמקרה של החזרת מטריצה, יש להחזיר מטריצה חדשה!

1. אתחול (כמובן) – בתהליך האתחול נוצרת ומאותחלת מטריצה 2×2 ע"י העברת ערכי D, C, B, A .
2. שחלוף (Transpose) – מחזירה את השחלוף של המטריצה. במטריצה 2×2 שחלוף הוא החלפה בין B ו- C . ומקבלים את המטריצה: $\begin{bmatrix} A & C \\ B & D \end{bmatrix}$. נסמן שחלוף כך M^T .
3. דטרמיננטה – מחזירה את הדטרמיננטה של המטריצה. (במטריצה 2×2 מוגדרת להיות $A \cdot D - B \cdot C$). נסמן דטרמיננטה כך $|M|$.
4. היפוך (Inverse) – מחזירה מטרימה שהיא ההיפוך של המטריצה. במטריצה 2×2 היפוך מוגדר כך: $\frac{1}{|M|} \cdot \begin{bmatrix} D & -B \\ -C & A \end{bmatrix}$. נסמן היפוך כך M^{-1} . אם ההיפוך לא קיים ($|M|=0$) תוחזר מטריצת היחידה $I_{2 \times 2}$.
5. מכפלת שתי מטריצות – הפונקציה מקבלת שתי מטריצות ומחזירה מטריצה 2×2 המכילה את המכפלה של שתי מטריצות (שהיוו קלט). מכפלה מוגדרת כך: $\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + GB & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$.
6. חלוקת שתי מטריצות – הפונקציה מקבלת שתי מטריצות ומחזירה מטריצה 2×2 המכילה את החלוקה של המטריצה הראשונה בשנייה. חלוקה של המטריצה N במטריצה M ניתנת לסימון כך: $N/M = N \cdot M^{-1}$.
7. חיבור שתי מטריצות – הפונקציה מקבלת שתי מטריצות ומחזירה מטריצה 2×2 המכילה את הסכום של שתי המטריצות.
8. חיסור שתי מטריצות – הפונקציה מקבלת שתי מטריצות ומחזירה מטריצה 2×2 המכילה את החיסור של השנייה מהראשונה.
9. כפל בסקלר – הפונקציה מקבלת מטריצה וסקלר (מסוג double) ומחזירה מטריצה 2×2 שמכילה את תוצאת הכפל בסקלר (פעולת כפל עם סקלר מוגדרת על כל אחד מאיברי המטריצה. למשל כפל בסקלר α ייתן את המטריצה הבאה: $(\alpha \cdot \begin{bmatrix} A & C \\ B & D \end{bmatrix}) = \begin{bmatrix} \alpha \cdot A & \alpha \cdot C \\ \alpha \cdot B & \alpha \cdot D \end{bmatrix}$).
10. הדפסה – מדפיסה את איברי המטריצה.
11. שחרור המטריצה – משחרר את הזיכרון שתפש מבנה הנתונים.

בשאלה זו אתם נדרשים לתת קובץ הצהרות (מנשק) וקובץ מימוש של ADT אשר עונה על הדרישות הנ"ל ועל הדרישות של תכנות במתודולוגית ADT כפי שנלמד בקורס. וכן לכתוב תכנית המשתמשת ב-ADT הנ"ל.

א. (8 נק') עליכם לתת קובץ מנשק (הצהרות) של ADT שעונה על הדרישות הנ"ל. עבור כל פעולה אריתמטית יש להגדיר פונקציה נפרדת. למשל חיבור שתי מטריצות ייקרא Add וכפל בסקלר ייקרא Mult_S.

ב. (9 נק') עליכם לממש את ה-ADT של מטריצה 2×2 . בקובץ המימוש יש להגדיר את מבנה הנתונים, ולממש אך ורק את 3 הפונקציות: אתחול, חלוקת שתי מטריצות, וכפל מטריצה בסקלר.

ג. (8 נק') עליכם לכתוב תכנית main שתחשב ותדפיס את תוצאת הביטוי: $2 \cdot E^T + F/G$. כאשר:

$$E = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad F = \begin{bmatrix} 0.35 & 7 \\ 8.7 & 6.89 \end{bmatrix}, \quad G = \begin{bmatrix} 1.5 & 1.2 \\ 0.3 & 1.1 \end{bmatrix}$$

הערה: במקרה יש יצירת מטריצות חדשות, יש להקפיד על שחרור הזכרון עם תום השימוש בהן !

שאלה מס' 2 (30 נק')

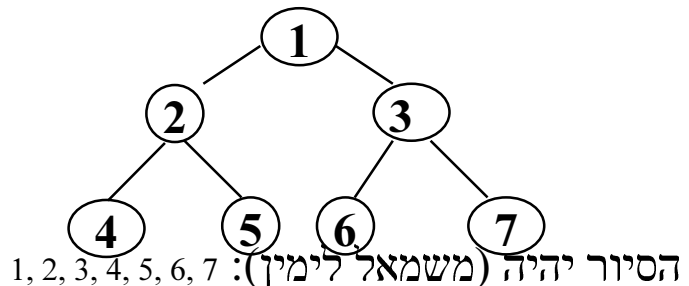
יש צורך במבנה נתונים שיחזיק איברים חיוביים שלמים. מספר האיברים שיש להחזיק חסום ב- N . (כלומר יש להחזיק לכל היותר N איברים במבנה). N זה מקיים $N = 2^0 + 2^1 + 2^2 \dots + 2^j$ עבור j ידוע.

עליך לממש עץ חיפוש בינארי שיאפשר את הפעולות הבאות:

- אתחול המבנה $O(N)$.
- הכנסת איבר חדש $O(\log N)$.
- חיפוש איבר $O(\log N)$.
- סיור inorder.
- סיור depth-first (*)

(*) סיור depth-first הינו סיור העובר על האיברים ע"פ רמתם בעץ ומשמאל לימין, איבר ברמה קרובה יותר לשורש יבוקר לפני כל האיברים ברמה מתחתיו.

לדוגמא: עבור העץ הנתון (לאו דווקא עץ חיפוש) :



שיים/לב: כל המספרים בעץ חיוביים. כמו כן ניתן להתייחס אל מבנה הנתונים כאל משתנה גלובלי.

- הסבר/י את מבנה הנתונים, וצרפ/י תרשים.
- כתוב/י קוד לפונקציה אתחול המבנה `init`.
- כתוב/י קוד לפונקציה ההכנסה `insert`. פונקציה זו תהיה רקורסיבית ! כמו כן כתבי את שורת הקריאה הראשונית לפונקציה הרקורסיבית (לדוגמא: בעץ נוח בד"כ לבצע קריאה זו אל שורש העץ). החלט/י כרצונך מה תחזיר הפונקציה ובאיזה תנאים, אך תעד/י זאת בקוד.
- כתוב/י קוד לפונקציה החיפוש `search`. פונקציה זו לא תהיה רקורסיבית !
- כתוב/י קוד לסיור inorder המדפיסה!! את האיברים תוך כדי הסיור. האם האיברים יודפסו בסדר מסוים?
- כתוב/י קוד לסיור depth-first המדפיסה!! את האיברים תוך כדי הסיור.

שאלה מס' 3 (25 נק')

נתונה המחלקה Range שתפקידה לבדוק האם מספר נמצא בתחום של מספרים שלמים. המחלקה מכילה שני ערכים – גבול תחתון (Low_) וגבול עליון (Up_). ניתן להניח כי תמיד מתקיים התנאי $Low_ \leq Up_$. נסמן תחום Range כך $[Low_ , Up_]$. לדוגמא $[5, 24]$.

קיימת מתודה IsInRange המקבלת מספר (מסוג float) ומחזירה true אם הוא נמצא בתחום (כולל הקצוות). המתודה IsEmpty מחזירה true אם התחום "ריק". תחום "ריק" מוגדר להיות $[0, -1]$ (שימו לב כי תחום זה אינו חוקי ...).

הקבועים MIN_INT ו-MAX_INT מוגדרים כערך המינימלי והמקסימלי (בהתאמה) שיכול להיות למספר שלם במחשב. (מבחינתנו הם מקבילים למינוס אינסוף ולפלוס אינסוף).

להלן הגדרת המחלקה:

```
// Range.H
#include <iostream.h>

class Range {
public :
    Range(int Lower=MIN_INT,int Upper=MAX_INT) :
        Low_(Lower),Up_(Upper) { }

    bool IsInRange(float num);

    print () { cout << "Lower Bound " << Low_ <<
        "Upper Bound " << Up_ << endl; }

    bool IsEmpty() { return Low_ == 0 && Up_ == -1 }
private :
    int Low_,Up_;
};
```

א. ממשו בקובץ Range.C את המתודה IsInRange כפי שהוגדרה.

ב. הוסיפו למחלקה (וממשו בקובץ Range.C) אופרטור חיתוך (האופרטור *) המקבל שני אובייקטים מסוג Range ומחזיר אובייקט חדש מסוג Range המכיל את חיתוכם. חיתוך של שני תחומים הוא התחום המשותף לשניהם. למשל נניח כי התחומים הם $[1, 10]$, $[5, 15]$ אזי החיתוך הוא $[5, 10]$. אם לא קיים תחום משותף (ז"א – התחומים זרים זה לזה), יוחזר תחום "ריק".

ג. הוסיפו (וממשו) למחלקה Range את המתודות GlobalLow/GlobalUp המחזירות את הגבולות העליון המקסימלי/התחתון המינימלי (בהתאמה) של כל התחומים שנוצרו עד עתה (גם כאלו שאינם קיימים יותר). אם לא נוצר עדיין אף תחום, יוחזר עבור גבול עליון מקסימלי הערך MIN_INT ועבור גבול תחתון מינימלי הערך MAX_INT.

7. אנו מעוניינים ליצור מחלקה חדשה הנקראת `OpenRange`. מחלקה זו מחזיקה בתוכה תחום "פתוח". ההבדל בין תחום "פתוח" לבין תחום "סגור" (`Range`) הוא שתחום פתוח לא כולל את הקצוות של עצמו. ז"א מספר שייך לתחום "פתוח" אם הוא נמצא בתחום, לא כולל הקצוות. נסמן תחום פתוח כך: `(Low_, Up_)`. לדוגמא: התחום הפתוח `(5,24)`. המספרים 5, 24 אינם שייכים לתחום זה, אך למשל 20.5 שייך לתחום זה. הגדירו וממשו את המחלקה `OpenRange` ע"י שימוש במנגנון הירושה. שימו לב כי עליכם לשנות תחילה את הצהרת המחלקה `Range` כך שתוכלו לרשת ממנה. בצעו תחילה את השינויים המתאימים במחלקה `Range`.

ה. נתון קטע הקוד הבא:

```
#include <iostream.h>
#include "Range.H"
#include "OpenRange.H"

void main() {
    Range *r1;
    r1=new OpenRange(1,20);
    if (r1->IsInRange(1))
        cout << "1 belongs to (1,20)" << endl;
    else
        cout << "1 does not belong to (1,20)" << endl;
}
```

1. בהינתן המימוש שלכם, מה יודפס ?
2. אם צריך – הסבירו בקצרה אילו/איזה שינויים צריך לבצע בקובץ/קבצי ההצהרות/המימושים של `Range` / `OpenRange` כך שהקוד יעבוד נכון (ייקבע כי 1 אינו בתחום הפתוח `(1,20)`). אם לא צריך לשנות כלום, נמקו.

שאלה מס' 4 (20 נק')

רשום script שתפקידו "לנקות" (למחוק) קבצים בספרייה ששמה /usr/adm/dumps לפי המתכונת הבאה:

בספרייה זו נמצאים קבצים השומרים אינפורמציה שוטפת בקשר לגיבויים. שם כל קובץ מכיל בתוכו את התאריך בו נוצר הקובץ.

דוגמאות לשמות קבצים :

/usr/adm/dumps/Jan11, /usr/adm/dumps/Feb15

כאשר מפעילים את ה-script עליו למחוק את כל הקבצים שנוצרו בחודש לפני הקודם. למשל, אם נפעיל את ה-script בחודש ינואר, עליו למחוק את כל הקבצים מחודש נובמבר.

הערות:

- אין לרשום בחירה ("case") של 12 אפשרויות, אנו מצפים ל-script חכם יותר...
- הנח שלמפעיל ה-script יש הרשאות כנדרש.
- דוגמא ל-output של פקודת date: Sun Jan 30 13:39:55 IST 2000