מבוא למדעי המחשב
202-1-1011

מבוא לתכנות למערכות מידע
202-1-1041

סמסטר א' תשס"ח
מבחן מועד ב'

פרופ' מיכאל קודיש
ד"ר מיכל זיו יוקלסון
גב' דיקלה דותן
מר שחר גולן
מר טל גרינשפון

משך המבחן: שלוש וחצי שעות.
חומר עזר אסור.

במבחן זה 5 שאלות. יש לענות על כולן. הניקוד מסתכם ב 100 נקודות.

אנא רשמו את תשובותיכם בטופס הבחינה בלבד. המחברת שקיבלתם הנה מחברת טיוטה בלבד ולא תימסר כלל לבדיקה. הקפידו לרשום על טופס הבחינה גם את מספר הנבחן וגם את מספר החדר בו אתם נבחנים.

מספר השורות העומדות לרשותכם בגוף השאלון רומז על אורך התשובה הנדרשת. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא.

<u>מותר להסתמך על סעיפים קודמים גם אם לא עניתם עליהם.</u>

בשאלות בהן לא מצוין אחרת, ניתן לבחור בפתרון רקורסיבי או פתרון שאינו רקורסיבי, לבחירתכם.

שימו לב: החשיבות העליונה היא לנכונות הקוד. מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה יעילה ומסוגננת תזכה בציון גבוה יותר.

בדף האחרון של המבחן ישנה תזכורת למספר פונקציות חשובות, בהן תוכלו להשתמש במהלך המבחן.

בהצלחה!!!

<u>שאלה 1 (25 נק')</u>

הממשק Filter מאפשר לנו לסנן איברים.

```
public interface Filter{
	public boolean accept(Object obj);
}
```

קריאה לשיטה accept(obj) תחזיר ערך בוליאני המסמן האם "לקבל" את העצם obj (במידה והערך המוחזר הוא true) או לא.

<u>סעיף א' (5 נק'):</u>

כתבו את כל המחלקה DivisibilityFilter המממשת את הממשק Filter. על הבונה במחלקה לקבל ערך מסוג int. על השיטה accept במחלקה לקבל רק עצמים מסוג Integer שערכם מתחלק בערך שקיבל הבונה. ניתן להניח שהקריאה ל-accept היא תמיד עם עצם מסוג Integer.

public class DivisibilityFilter implements Filter {

```
	int d;
	public DivisibilityFilter(int d){
		this.d=d;
	}
	public boolean accept(Object data) {
		return ((Integer)data).intValue()%d==0;
	}
}
```

<u>סעיף ב' (7 נק'):</u>

כתבו את כל המחלקה MajorityFilter המממשת את הממשק Filter. הבונה במחלקה מקבל שלושה עצמים מסוג Filter. השיטה accept של המחלקה תקבל אך ורק איברים אותם מקבלים לפחות שניים מהפילטרים שהתקבלו בבונה.

public class MajorityFilter implements Filter {

```
	Filter f1,f2,f3;
	public MajorityFilter(Filter f1,Filter f2,Filter f3){
		this.f1=f1;
		this.f2=f2;
		this.f3=f3;
	}
	public boolean accept(Object data) {
		return  (f1.accept(data)&&f2.accept(data))||
			(f1.accept(data)&&f3.accept(data))||
			(f2.accept(data)&&f3.accept(data));
	}
}
```

עבור הסעיפים ג-ד נתונה המחלקה List המכילה שיטה

```
public void addLast(Object data)
```

המוסיפה את האובייקט data לחוליה חדשה בסוף הרשימה. (אין צורך לממשה)

<u>סעיף ג' (8 נק'):</u>

השלימו את הפונקציה הבאה המקבלת Filter ו Iterable ומחזירה רשימה **<u>חדשה</u>** המכילה אך ורק את האיברים מתוך ה Iterable שהפילטר מקבל.

```
public static List filter(Iterable ds, Filter f){

      List ans=new List();
      for (Object data : ds)
            if (f.accept(data))
                  ans.addLast(data);
      return ans;
}
```

<u>סעיף ד' (5 נק'):</u>

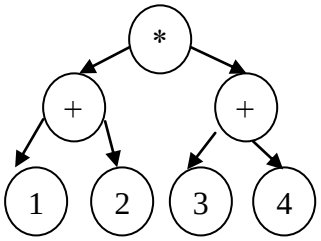
השלימו את הפונקציה הבאה המקבלת שלושה מספרים מסוג int ומחזירה רשימה **<u>חדשה</u>** המכילה אך ורק את האיברים מסוג Integer שמתחלקים לפחות בשניים מהמספרים.

```
public static List filter(Iterable ds, int d1, int d2, int d3){
      return filter(ds,new MajorityFilter(
                              new DivisibilityFilter(d1),
                              new DivisibilityFilter(d2),
                              new DivisibilityFilter(d3)));
}
```

<u>שאלה 2 (20 נק')</u>

ניתן לייצג ביטוי חשבוני באמצעות עץ בינארי מלא שבצמתיו יש אובייקטים מסוג `String`. עץ כזה ייקרא "עץ חשבוני".

לדוגמא:



הביטוי ((1+2)*(3+4)) ייוצג ע"י

שימו לב כי סדר הפעולות מוגדר באופן יחיד ע"י העץ. הפעולה בשורש מבוצעת לאחר חישוב הביטויים בתתי העצים שלו.
בשאלה זו נניח כי הקלט הנתון הינו מחרוזות ללא רווחים בהן המספרים הם שלמים בעלי ספרה אחת בלבד.

<u>סעיף א' (10 נק'):</u>

כתוב שיטה המקבלת שורש של עץ חשבוני המייצג ביטוי חשבוני. השיטה תחזיר מחרוזת המייצגת את אותו ביטוי חשבוני עם סוגריים המגדירים באופן מלא את סדר הפעולות.

```
public static String getExpression(BNode root){
  if (root==null) return "";
  if (root.left==null)
       return root.data.toString();
  return "("+getExpression(root.left)
              +root.data
                 +getExpression(root.right)+")";
}
```

סעיף ב' (10 נק'):

כתוב שיטה המקבלת מחרוזת המייצגת ביטוי חשבוני עם סוגריים המגדירים באופן מלא את סדר הפעולות. השיטה תחזיר שורש של עץ חשבוני המייצג את אותו ביטוי חשבוני.

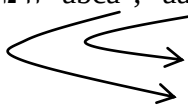
```
public static BNode getRoot(String exp){
   Stack s=new StackAsArray();
    for (int i=0;i<exp.length();i++)
       if (exp.charAt(i)==')'){
           BNode op1=(BNode)s.pop();
           BNode root=(BNode)s.pop();
           root.right=op1;
           root.left=(BNode)s.pop();
           s.push(root);
       }
       else if (exp.charAt(i) !='(')
          s.push(new BNode(new String(exp.charAt(i))));

  return (BNode)s.pop();
}
```

<u>שאלה 3 (20 נק')</u>

נגדיר דמיון בין שתי מחרוזות כמספר האותיות הזהות (בסדרן המקורי) המקסימלי המופיע בשתי המחרוזות.

על מנת לחשב מספר זה, יש "להעמיד" את המחרוזות זו על גבי זו כדי ליצור התאמה, באופן הבא: אותיות יועמדו זו על גבי זו רק במידה והן זהות. לא ניתן לשנות את סדר האותיות במחרוזות, אך ניתן לרווח את המחרוזת כדי ליצור את ההתאמה. הדמיון יתקבל מההעמדה האופטימלית: ההעמדה בה מספר האותיות התואמות הינו רב ביותר.

למשל, הדמיון בין המחרוזות "aacdd", "abca" הינו 2, משום שההעמדה אופטימלית הינה:



a	a		c		d	d
a		b	c	a		

תיתכן יותר מהעמדה אופטימלית אחת. למשל, ניתן להעמיד את המחרוזות "aacdd", "abca" גם באופנים הבאים:

a	a		c		d	d
	a	b	c	a		

a			a	c	d	d
a	b	c	a			

אלו הן העמדות אופטימליות משום שבכולן יש שתי אותיות ממחרוזת אחת ה"עומדות" על אותיות זהות במחרוזת השניה. מידת הדמיון של שתי המחרוזות הנ"ל הינה 2, משום שלא קיימת העמדה המאפשרת 3 התאמות. במידה ולשתי המחרוזות אין אף אות משותפת, מידת הדמיון ביניהן היא 0.

השלימו את הפונקציה <u>הרקורסיבית</u> alignStrings המקבלת כקלט שתי מחרוזות ומחזירה את מידת הדמיון ביניהן. ניתן להניח שמחרוזות הקלט אינן null.

```
public static int  allignStrings(String s1, String s2) {
   int ans;
   if (s1.length()==0 | s2.length()==0) {
       ans = 0;
   }
   else {
      if (s1.charAt(0) == s2.charAt(0)) {
          ans = 1+allignStrings(s1.substring(1),
                                s2.substring(1));
      }
      else {
         ans=Math.max(allignStrings(s1,s2.substring(1)),
                      allignStrings(s1.substring(1),s2));
      }
   }
   return ans;
}
```

שאלה 4 (20 נק')

השאלות הבאות מתייחסות לקטע הקוד המופיע בסוף השאלה (בדף הבא).
בכל סעיף (4 נק') תשובה נכונה אחת. הקיפו את מספר הסעיף בדפי המבחן.

סעיף 1

1) לאחר ביצוע הפונקציה insert1, איברי המערך במקומות array[0] .. array[i] יהיו ממוינים, בתנאי שלפני הקריאה לפונקציה איברי המערך array[0] .. array[i-1] היו ממוינים בסדר עולה.

2) לאחר ביצוע הפונקציה insert1, איברי המערך במקומות array[0] .. array[i] יהיו ממוינים, בתנאי שלפני הקריאה לפונקציה, איברי המערך array[0] .. array[i-1] היו ממוינים בסדר יורד.

3) לאחר ביצוע הפונקציה insert1, איברי המערך במקומות array[0] .. array[i] יהיו ממוינים. המקטע כולו ימויין תוך כדי ביצוע הפונקציה.

4) מטרת הפונקציה insert1 זהה למטרת הפונקציה insert2.

סעיף 2

1) תתכן שגיאה בזמן ריצה מכיוון שלפונקציות insert1 ו- insert2 חתימות זהות.

2) הקוד הנ"ל שגוי (לא יעבור קומפילציה), משום שלפונקציות insert1 ו- insert2 חתימות זהות.

3) רק קריאה לפונקציה what עם מערך שהינו null, תגרור שגיאת קומפילציה.

4) קריאה לפונקציה what עם מערך שהינו null, תגרור שגיאה בזמן ריצה.

סעיף 3

1) בקריאות Math.max(a,b) ,Math.min(a,b) מעורבים שני מופעים של המחלקה Math.

2) החלפת האופרטור && באופרטור &, בכל אחד משני המקומות בהם מופיע "&&" בפונקציה insert2, לא תשנה את תוצאת ריצת קטע הקוד.

3) לא ניתן לקרוא לפונקציות insert1, insert2 מתוך הפונקציה what, משום שלא ניתן לקרוא לפונקציות שהן private מפונקציה שהינה public.

4) החלפת שורות 3-6 בשורות הבאות:

```
for (i = 1; i < array.length-1; i = i+2)
    insert2(array, i+1);
```

לא תשנה את ריצת הפונקצית what .

סעיף 4

1) הפעלת הפונקציה what ממיינת מערך, בסדר עולה.

2) הפעלת הפונקציה what ממיינת מערך, רק במידה ואורכו זוגי.

3) הפעלת הפונקציה what ממיינת מערך, רק במידה ואורכו אי-זוגי.

4) קיימים מערכים באורך זוגי וכן מערכים באורך אי-זוגי שלא ימוינו ע"י הפעלת הפונקציה what.

סעיף 5 (ניתוח זמן ריצה במקרה הגרוע)

1) זמן הריצה של קטע הקוד הנ"ל, על מערך באורך n הינו בסדר גודל של n פעולות.

2) זמן הריצה של קטע הקוד הנ"ל, על מערך באורך n הינו בסדר גודל של n^2 פעולות.

3) זמן הריצה של קטע הקוד הנ"ל, על מערך באורך n הינו בסדר גודל של log(n) פעולות.

4) זמן הריצה של קטע הקוד הנ"ל, על מערך באורך n הינו בסדר גודל של n*log(n) פעולות.

שאלה 5 (15 נק')

נתונים הממשק והמחלקות:

```
public interface I {}
public abstract class A { private int a = 1; }
public abstract class B extends A { public static int b = 2; }
public class C extends A implements I { private int a=3; }
public class D extends C {}
public class E extends C {}
public class F extends B {}
public class G extends F { public int b=4; }
```

המחלקות D, E ו-F הינן ריקות, כלומר ללא שדות או שיטות. באופן דומה הממשק I הינו ממשק ריק.
כמו כן, נתון קטע הקוד הבא:

```
public static void main(String [] args) {
    C c = new C();
    D d = new D();
    E e = new E();
    F f = new F();
    G g = new G();

// הוספת השורה בכל סעיף הינה במקום זה
}
```

לגבי כל אחת משורות הקוד הבאות (1-15), סמנו מה יקרה אם נכתוב אותה במקום שורת ההערה ב-main – האם תהיה שגיאת קומפילציה, האם התוכנית תיתקל בשגיאה (חריגה) בזמן ריצה, או שהקוד ירוץ בצורה תקינה. במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה (בסעיפים 1-15) **יש לציין את הסיבה**.

שימו לב: הסעיפים הינם בלתי-תלויים.

במקרה של ריצה תקינה של הקוד בסעיפים 11-15 (הכוללים System.out.println) יש לכתוב את הערך שיודפס למסך.

1) e = c;

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

טיפוסים לא מתאימים

2) e = (E)c;

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

ClassCastException: C cannot be cast to B

3) c = e;

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

---

4) c = (C)e;

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

---

5) d = e;

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

טיפוסים לא מתאימים

6) d = (D)e;

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

טיפוסים שאינם ניתנים להמרה

7) A ab = new B();

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

לא ניתן ליצור מופע של מחלקה אבסטרקטית

8) A af = new G();

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

---

9) C ci = new I();

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

לא ניתן ליצור מופע של ממשק

10) I ic = new C();

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה בלבד):

---

11) System.out.println(c.a);

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה) או את הפלט (ריצה תקינה):

השדה a הוא פרטי במחלקה C

12) System.out.println(g.b);

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה) או את הפלט (ריצה תקינה):

4

13) System.out.println(((B)g).b);

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה) או את הפלט (ריצה תקינה):

2

14) System.out.println(((G)f).b);

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה) או את הפלט (ריצה תקינה):

ClassCastException: F cannot be cast to G

15) System.out.println(B.b);

סמנו מה יקרה: שגיאת קומפילציה / שגיאה בזמן ריצה / ריצה תקינה

ציינו את הסיבה (במקרה של שגיאת קומפילציה או שגיאה בזמן ריצה) או את הפלט (ריצה תקינה):

2