**מבוא לחישוב - שאלות דוגמא לבחינה**

מסמך זה מכיל שאלות שונות ברוח החומר של ״מבוא לחישוב״ שמתאימות להכנה בחינה, השאלות אינן בפורמט של בחינה אלא בעיקר נועדו להיות אוסף מקיף של דוגמאות לשאלות לתרגול עצמי.

חומר העזר שיצורף לבחינה נמצא בקישר הזה:

<https://github.com/benmoshe/Intro2CS/tree/main/Exams>

כל דוגמאות הקוד של הקורס, נמצאות בקישור הזה:

<https://github.com/benmoshe/Intro2CS/blob/main/src>

להלן קיושר לאתר עם בחינות (ופתרונות חלקיים) של בחינות משנים עברו - שימו לב שבעבר הבחינות כיסו פחות חומר והשאלות היו קלות יחסית

<https://github.com/LeviEyal/JAVACourse-TestsSolutions>

**שאלה 1:**

כתבו פונקציה המקבלת שני מספרים שלמים אי שליליים n,m ומחזירים מטריצה שכל תא שלה מייצג את מספר הדרכים **המונוטוניות** האפשרויות להגיע מ(0,0) לאותו התא.

לדוגמא:

עבור n=2,m=3

הפלט יהיה:

1 1 1

3 2 1

חתימת הפונקציה:

public static int[][] Q1(int n, int m)

**שאלה 2:**

כתבו פונקציה רקורסיבית בשם q2 שמקבלת מחרוזת ומחזירה אותה "מצומצמת".  
כל רצף של אותיות זהות הופך לאות אחת.

לדוגמה עבור הקלט : “aaabbbcccxxyaaa” הפלט יהיה “abcxya”

חתימת הפונקציה:

public static String q2(String str){

**שאלה 3:**

כתבו פונקציה שמקבלת שתי מחרוזות s1,s2 ומחזירה מספר מופעים של מחרוזות s2 בתוך s1.

לדוגמה עבור הקלט: "s1=“to **be** or not to **be**”, s2=”be” הפלט יהיה 2. שימו לב שיש להתייחס למספר מופעים עם חזורת, משמע גם עבור הקלט מטה התשובה תהיה 2.

s1=“aaa”, s2=”aa"

חתימת הפונקציה:

public static int Q5(String s1, String s2)

**שאלה 4-6:**

בשאלות מטה נעסוק ברשימות מקושרות (כפי שמוגדרת בממשק MyListInterface בחומר העזר) , הניחו שקיימת לכם המחלקה MyList<T> l, שמממשת את הממשק MyListInterface, ובעלת בנאי ריק שיוצר רשימה ריקה.

**שאלה 4:**

3.1 כתבו פונקציה סטטית שמקבלת רשימה מקושרת ומחזירה אמת אך ורק אם **כל** איברייה **שונים** זה מזה (משמע אם איברי הרשימה מייצגים קבוצה), אחרת שקר.

public static boolean isSet(MyListInterface<T> l) {//…

3.2 כתבו פנוקציה סטטית שמקבל רשימה מקושרת ומחזירה רשימה חדשה שמייצגת את קבוצת האיברים ברשימה (עותק יחיד לכל איבר):

public static MyListInterface<T> toSet(MyListInterface<T> l) {//…

**שאלה 4 (בסגנון אחר):**

4.1 כתבו פנוקציה סטטית שמקבל שתי רשימות מקושרת ומחזירה רשימה חדשה שמייצגת את קבוצת החיתוך של שתי רשימות הקלט (עותק יחיד לכל איבר):

public static MyListInterface<T> intersect(MyListInterface<T> l1, MyListInterface<T> l2) {//…

4.2 כתבו פנוקציה סטטית שמקבל שתי רשימות מקושרת ומחזירה רשימה חדשה שמייצגת את קבוצת האיחוד של שתי רשימות הקלט (עותק יחיד לכל איבר):

public static MyListInterface<T> union(MyListInterface<T> l1, MyListInterface<T> l2) {//…

**שאלה 5:**

כתבו פונקציה שמקבלת רשימה מקושרות ומחזירה את כמות המחלקות השונות אליהן שייכים האובייקטים ברשימה, לדוגמא אם המרשימה מכילה רק אובייקטים מסוג String יוחזר 1. הדרכה לכל מחלקה יש שיטה getClass אשר מחזירה את המחלקה אליו שייך האובייקט ולכל מחלקה יש שיטה getSimpleName שמחזירה מחרוזת עם שם המחלקה אליו שייך האובייקט (הטיפוס המוצבע בזמן אמת)

public static int Q7(ListInterface <T> l) {//…

**שאלה 6:**

מספר ספניק(sphenic number) הוא מכפלה של שלושה מספרים ראשוניים שונים. כלומר מספר ספניק מקיים כאשר וגם כולם ראשוניים. לדוגמה 30 הוא מספר ספניק שכן אבל 98 אינו מספר ספניק שכן ו-15 אינו מספר ספניק שכן ו-1 אינו נחשב לראשוני.

כתבו פונקציה המקבלת מספר ומחזירה אמת אם הוא מספר ספניק, אחרת שקר.

חתימת הפונקציה

public static Boolean Q10(int n)

**שאלה 7:**

כתבו פונקציה שמקבלת מערך של מספרים ממשיים ומחזירה מערך חדש שמכיל את תת המערך המונוטוני עולה הארוך ביותר. למשל עבור [4,2,3,5,0,1] הפונקציה תחזיר [2,3,5] בעוד שעבור המערך [2,1, 3] הפונקציה תחזיר מערך באורך 1 למשל [2] או [1] או [3].

ניתן להניח כי איברי המערך שונים זה מזה.

חתימת הפונקציה:

public static int[] Q12(double[] arr)

**שאלות 8-11: שאלות על עצים:**

שאלות אילו מתייחסות לממשק של עצים בינאריים BinaryTree<T> כפי שמופיעה בחומר העזר.

**שאלה 8:**

בשאלה זו נתייחס רק למבנה העץ ולא לתוכנו, כתבו פונקציה המקבלת שני עצים בינאריים ומחזירה אמת אם ורק אם הם בעלי אותה צורה מבחינת מבנה הצמתים (ללא קשר לערכי המידע בקודקודים):

8.1 כתבו (בעברית) - את ההגדרה הרקורסיבית לשוויין מבני בין שני העצים.

8.2 השלימו את הפונקציה:

public static boolean isOfTheSameStructure(BinaryTree bt1, BinaryTree bt2) //{...}

8.3 כתבו פונקציית בדיקה לפונקציה מעלה:

@Test

public void testIsOfTheSameStructure() //{...}

**שאלה 9:**

כתבו פונקציה שמחשבת כמה **עלים** יש בעץ בין הרמה ה min (כולל) לרמה ה max (לא כולל). שימו לב השורש הוא ברמה ה 0.

public static <T> int q9(BinaryTree<T> bt1, int min, int max);

**שאלה 10:**

כתבו פונקציה שמחזירה ArrayList של כל העלים בעץ (בסדר InOrder):

public static <T> ArrayList<T> q10(BinaryTree<T> bt1)

**שאלה 11:**

כתבו פונקציה שמקבלת עץ בינארי ומחזיקה את יחס הקומפטיות של העץ, נגדיר את יחס הקומפקטיות של העץ להיות היחס בין גובה העץ לבין הגובה המינימאלי של עץ בינארי שמכיל את כל קודקודי העץ.

הדרכה: היחס צריך להיות מספר ממשי גדול או שווה ל1. עץ בעל 7 קודקודים יכול להיות בגובה 2,3,4,5,6 אם הוא בגובה 2, היחס יהיה 1, אם הוא גבוה 6 (מקסימאלי) היחס יהיה 3.

הדרכה: ניתן להניח שהעץ מכיל יותר מקודקוד יחיד.

public static double q10(BinaryTree bt){

**שאלה 12:**

כתבו פונקציה שמקבלת מחרוזת של אותיות s ומחזירה מחרוזת אשר מופיעים בה רק אותן אותיות המופיעות פעם אחת ב-s. סדר האותיות במחרוזת החדשה יהיה זהה לסדרם ב-s. אם אין אות המופיעה בדיוק פעם אחת, החזירו מחרוזת ריקה.

דגשים:

* במחרוזת s יש רק אותיות קטנות מהabc
* הפונקציה צריכה להיות יעילה, לכן מותר לעבור על מחרוזת s רק פעמיים. שימו לב שכל קריאה לפונקציות contains או IndexOf עלולה לגרום למעבר על כל המחרוזת s.

דוגמאות:

עבור הקלט “abcdcb” הפלט יהיה “ad”

עבור הקלט “abracadabra”ה הפלט יהיה “cd”

חתימת הפונקציה:

public static String Q12(String s)

**שאלה 13 :**

כתבו פונקציה המקבלת מערך של אובייקטים וקומפרטור וממיינת את המערך לפי הקומפורטור (בסדר עולה) על הפונקציה שלכם להיות יעילה ככול הניתן,

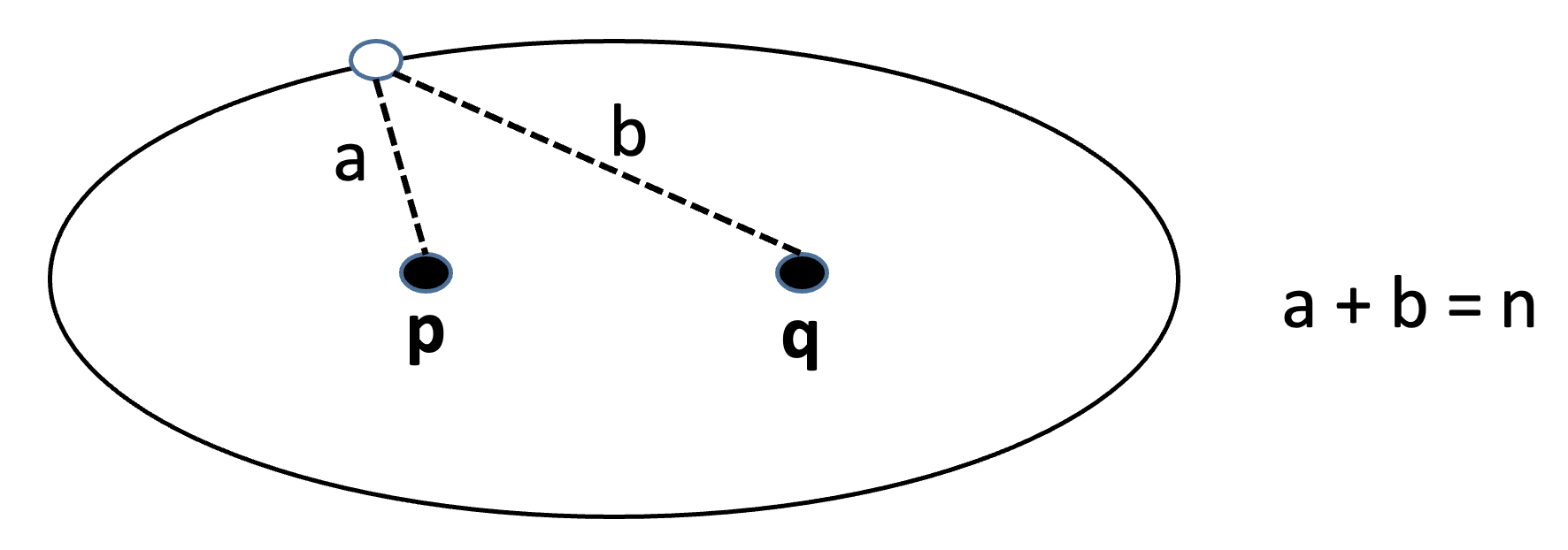
חתימת הפונקציה:

public static<T> void mySort(T[] p, Comparator<T> comp)

כתבו מחלקת בדיקה אשר משווה את ביצועי הפונקציה שלכם לזו של המיון של Arrays.sort הבדיקה צריכה להתבצע על מערכים מעורבבים של מחרוזות בגודלים של 100,1000,10000.

**שאלה 14:**

כתבו מחלקה בשם **Ellipse** (אליפסה). האליפסה מוגדרת על ידי שתי נקודות מרכז p,q ומספר ממשי n. הקוד לנקודה נמצא בנספח, וחובה להתייחס רק אליו ללא שינויים.



1. כתבו את כותרת המחלקה ואת השדות (המשתנים) שלה.

2. כתבו שני בנאים: בנאי שמקבל חמישה נתונים **בסדר** (משמאל לימין) xp,yp,xq,yq,n, ובנאי מעתיק שמקבל אליפסה חדשה. בנאי המעתק ישתמש בהעתקה עמוקה.

3. הוסיפו למחלקה שיטה שמקבלת נקודה ומחזירה 0 אם הנקודה נמצאת על שפת האליפסה, -1 אם היא נמצאת בתוך האליפסה, או 1 אם היא נמצאת מחוץ לאליפסה. נקודה נמצאת על שפת האליפסה אם סכום של המרחק שלה משתי נקודות המרכז שווה ל-n. אם הסכום יותר מ-n, אזי היא מחוץ לאליפסה, ואם הסכום פחות מ-n אזי הוא בפנים (באיור למעלה המרחקים לנקודה **r** הם a,b).

**{...}public int where(Point r)**

4. הוסיפו למחלקה שיטה להשוואת שתי אליפסות. (שני אליפסות שוות אם הנקודות שלהם זהות, וגם הערכי ה-n שלהן זהים.) השיטה מחזירה true אם האליפסות שוות, אחרת היא מחזירה false.

**{...}public boolean equals(Ellipse e)**

**שאלה 15**

בשאלה זו נתייחס לממשק Parabula שמייצג משוואה ריבועית מהצורה ax^2+bx+c=0:

interface Parabula {

double f(double x); // computes the value of this parabula at x.

Parabula add(Parabula p); // computes a new Parabula = p + “this”.

double[] get(); // retunrs an array of doubles {a,b,c}: ax^2+bx+c=0;

double extream(Parabula p) // returns the x value of the extrema point

//(min or max), if none throws an Exception.

}

כתבו את המחלקה F2 **שמממשת** את הממשק Parabula ומייצגת משוואה ריבועית, מהצורה: ax^2 + bx + c =0. הדרכה: למחלקה יש בנאי שמקבל שלושה ממשיים a,b,c.

**שאלה 16**

בשאלה זו נתייחס לממשק Parabula כפי שמופיע מעלה (שאלה 15), ממשו את הפונקציות הסטטיות הבאות:

4.1 (10 נקודות) בהינתן פרבולה, כתבו פונקציה סטטית שמחזירה כמה שורשים **ממשיים** **שונים** קיימים לפרבולה:

static int numberOfRealRoots(Parabula p) {…}

4.2 (15 נקודות) כתבו פונקציה סטטית שמקבלת אוסף של Parabula וממיינת אותו לפי ערכי ה y של נקודות הקיצון (בסדר יורד – מגדול לקטן).

static void sort(ArrayList<Parabula> a) {…}

**שאלה 17**

בשאלה זו התייחסוו לממשק של צורה גיאומטרית כפי שמופיעה בחומר העזר GeoShape:

נתון הממשק של Filter שמייצג מסנן של צורות,

public interface ShapeFilter{

public boolean filter(GeoShape s); // returns true iff s passes this Filter.

}

הממשק פילטר של צורות כולל פונקציה יחידה שמקבלת צורה ומחזירה אמת או שקר. אם הפילטר מחזיר ״אמת״ נגיד שהצורה ״עברה״ את הפילטר ואם הפילטר מחזיר ״שקר״ נאמר שהצורה ״נכשלה״ בפילטר.

הדרכה: בשאלה זו ניתן להניח שהפרמטרים לבנאים תקינים - משמע אינם null.

1. השלימו את המחלקה AreaFilter שמממשת את הממשק ShapeFilter, ומקבלת בבנאי שלה צורה. הפילטר מעביר (מחזיר אמת) אך ורק צורות ששטחן גדול ממש מהשטח של הצורה בפילטר.

2. השלימו את המחלקה PointsFilter שמממשת את הממשק ShapeFilter, ומקבלת בבנאי שלה מערך של נקודות (אפשר להניח קלט תקין - משמע המערך אינו null, וכל תא בו מכיל נקודה שאינה null). הפילטר מעביר אך ורק צורות שמכילות את **כל** הנקודות במערך של הפילטר.

3. בהינתן מערך של צורות ופילטר השלימו הפונקציה הסטטית שמחזירה מערך חדש שמכיל אך ורק צורות שעברו את הפילטר, הדרכה: שימו לב שמערך התשובה צריך להיות בלתי תלוי בזיכרון של מערך הקלט.

public static GeoShape[] ff(GeoShape[] arr, Filter f){...}

**שאלה 18:**

הניחו שיש ״כספת״ שיש לה ״קודן״ עם בדיוק שישה תווים: 1,2,3,4,5,# , הקוד החוקי לכספת כולל רצף של 4 ספרות ללא חזרות, ואחריהן ״#״. כתבו פונקציה שמחזירה מערך מחרוזות שמייצג את כלל הקודים האפשריים של הכספת (ללא חזרות).

public static String[] allCodes(){//…

הקודים הבאים הם חוקיים: ״1234#״, ״5431#״,"1325#"

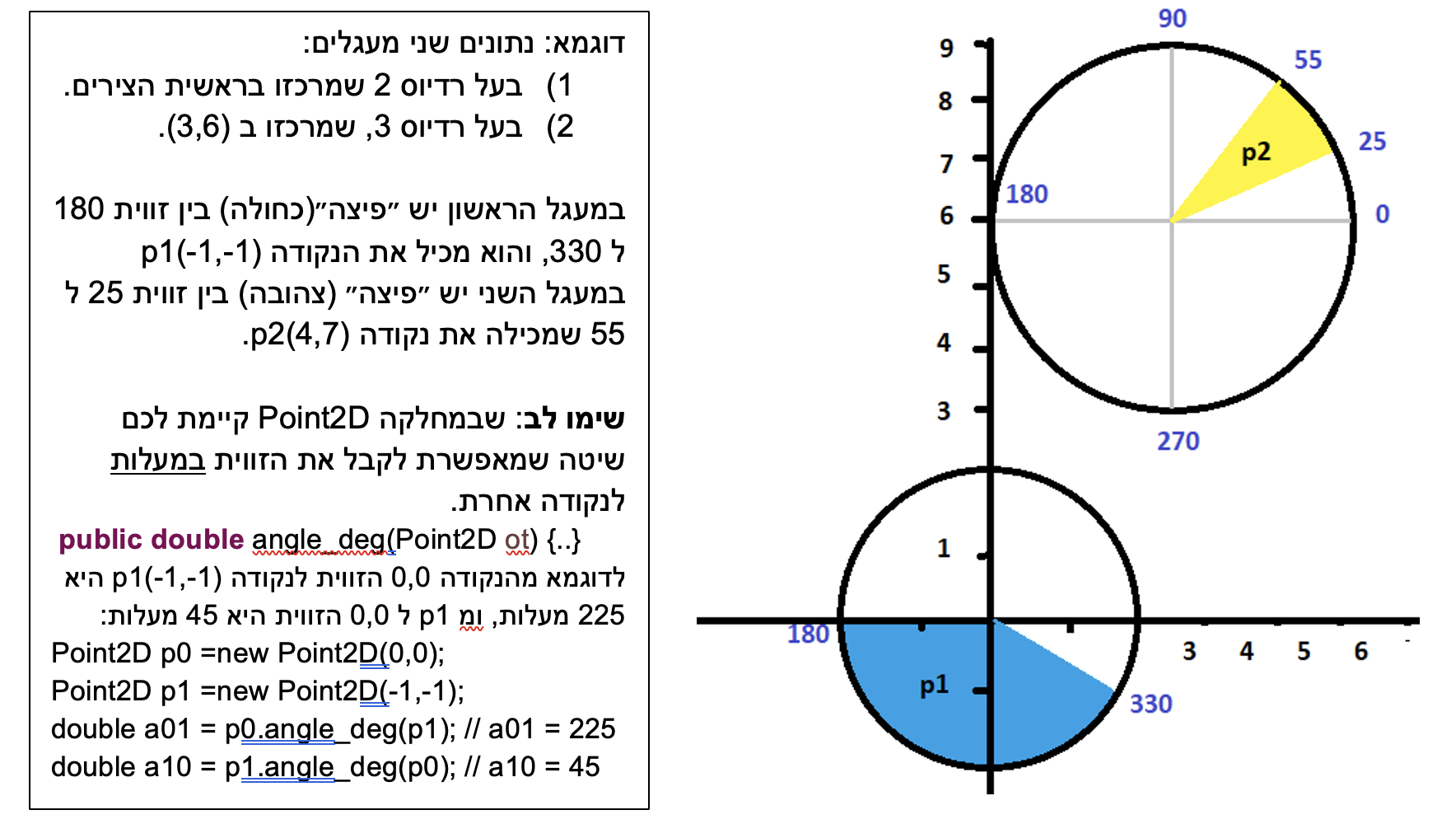
הקודים הבאים אינם חוקיים: ״2324#״ֿ, ״023#״, ״1237#״, ״12#34״

3.2 (5 נקודות) השלימו את פונקציית הבדיקה testAllCodes

**שאלה 19:**

בשאלה זו נכתוב את המחלקה גזרה (Pizza) שמייצגת חלק זוויתי של מעגל ("משולש פיצה"), בין זווית התחלתית לזווית סופית. ניתן להניח שהזווית הראשונה קטנה ממש מהזווית השנייה (שתיהן מוגדרות במעלות).

המחלקה מממשת את הממשק GeoShape (ראו קוד מצורף)



1. השלימו את הבנאי של המחלקה Pizza שמקבל מעגל, זווית התחלתית וזווית סופית.
2. השלימו את השיטות הנדרשות במחלקה Pizza לפי הממשק GeoShape:

**public** **interface** GeoShape {

**public** **boolean** contains(Point2D ot); // מחזיר אמת אם ורק אם הנקודה נמצאת בתוך הצורה - או על הגבול

**public** **double** area(); // מחזיר את השטח של הצורה

**public** **double** perimeter(); // מחזיר את ההיקף של הצורה

}

הדרכה: בשאלה זו אינכם נדרשים לעשות בדיקות - אבל הקוד שלכם צריך להיות נכון!