GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2022 מועד א'
סמל השאלון: 97104
קובץ: תשובות סופיות

אלגוריתמיקה ותכנות (Java ו-C#)

הנדסת תוכנה · הנדסאים וטכנאים

הוראות

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: **בשאלון זה שני מבחנים, עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:** מבחן ב- Java (עמוד 2) מבחן ב- C# (עמוד 15) בכל מבחן 14 שאלות. **חלק א' – 40 נקודות** שאלות 1-6: יש לענות על **ארבע** שאלות בלבד. ערך כל שאלה – 10 נקודות. **חלק ב' – 24 נקודות** שאלות 7-10: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 12 נקודות **חלק ג' – 36 נקודות** שאלות 11-14: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 18 נקודות. **בסך הכול: 100 נקודות.**

ג. חומר עזר מותר בשימוש: 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית). 2. קלסר **אחד בלבד** עם **חומר ההרצאות**. אין להוציא דפים מהקלסר. **אין לצרף ספרים או חוברות עם פתרונות.**

ד. <u>הוראות מיוחדות</u>: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת הבחינה). 3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות. 4. יש לכתוב את התשובות במחברת הבחינה **בעט בלבד**, בכתב יד ברור. 5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה. 6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק. 7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד. 8. יש להסביר בפירוט כל תוכנית שנכתבה, **תוכנית ללא הסבר מפורט לא תזכה בניקוד.** 9. אם לדעתכם חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לכם להמשיך בפתרון השאלה. נמקו את בחירתכם. **הנחיות כלליות לנבחנים - מבחן ב-JAVA:** 1 בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה: `Scanner in=new Scanner(System.in);` 2 דוגמה להוראה לקליטת מספר שלם: `int x = in.nextInt();` 3 דוגמה להוראה לקליטת מספר עשרוני: `double y = in.nextDouble();` 4 דוגמה להוראה לקליטת מחרוזת: `String str = in.next();` 5 הוראות לפלט על המסך: `System.out.print();` 6 יצירת מספר שלם אקראי num בין X ל-Y כולל (X <= num <= Y) שיטה א' - שימוש במחלקת שירות Math: `int num = X + (int)(Math.random()*(Y - X +1));` שיטה ב' - שימוש במחלקה Random: `Random rand = new Random();` `int num = X + rand.nextInt(Y - X + 1);` 7 מחזירה תת-מחרוזת ממקום k עד סוף המחרוזת. **הנחיות כלליות לנבחנים - מבחן ב-C#:** 1 דוגמה להוראה לקליטת מספר שלם: `int x = int.Parse(Console.ReadLine());` `int x =` 2 `Convert.ToInt32(Console.ReadLine());` דוגמה להוראה לקליטת מספר עשרוני: `double y =` 3 `double.Parse(Console.ReadLine());` דוגמה להוראה לקליטת מחרוזת: `string str =` 4 `Console.ReadLine();` הוראות לפלט על המסך: `Console.WriteLine();` 5 יצירת מספר שלם אקראי num בין X ל-Y כולל (X <= num <= Y): `int num =` `Random rand = new Random();` 6 `rand.Next(X, Y + 1);` הפונקציה `s.Substring(k)` מחזירה תת-מחרוזת ממקום k עד סוף המחרוזת.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

תשובות סופיות

שאלה 1

לולאת while שקולטת מחרוזות כל עוד המחרוזת אינה מתחילה ומסתיימת ב-'A'; מונה גדל כאשר התו הראשון או האחרון הוא 'Z'; בסוף מדפיסים את המונה.

שאלה 2

א. אחרי הזימון: arr = {1, 3, 10, 12, 17, 18}.

ב. למשל num = 100 (כל num גדול מ-18, או כל num קטן מ-1, למשל 0).

ג. מסדרת מחדש את המערך כך שהאיברים הקטנים מ-num נמצאים בתחילתו והאיברים הגדולים מ-num בסופו (חלוקה סביב num).

שאלה 3

א. מעבר אחד על המערך: אם יש 0 מחזירים false; אחרת סופרים חיוביים ושליליים ומחזירים countPos == countNeg.

ב. $O(n)$, כאשר n הוא אורך המערך.

שאלה 4

א. עוברים על המערך; לילד שנולד בשנה year מגדילים מונה בנים ('gender == 'M) או מונה בנות; בסוף מדפיסים את שני המונים.

ב. שני מערכי מונים בגודל 12 (בנים ובנות); לכל ילד מגדילים את התא month-1 במערך המתאים; בסוף מדפיסים 12 שורות.

שאלה 5

א. (1) `public class Movie` עם התכונות `private String name; private double price; private int num;` (ב-`string name`: C#`).

(2) `public Movie()` ו-`public Movie(String name, double price, int num)` (ב-`string C#: `name`).

ב. Java: 3000.0 ואחריו 1000.0 (ב-C#: 3000 ואחריו 1000).

ג. חיפוש מקסימום: שומרים את מקום הסרט עם הכי הרבה כרטיסים (getNum()) ובסוף מחזירים את getName() שלו.

שאלה 6

א. הפעולה מחזירה false.

ב. מחזירה true אם ורק אם למספרים מספר ספרות שונה והספרה השמאלית (הראשונה) שלהם שווה; אחרת false.

ג. מחזירה true אם למספרים מספר ספרות שונה, ו-false אם מספר הספרות שווה. דוגמה ל-false: where(123, 456).

שאלה 7

א. קולטים מספרים עד 0; לכל מספר num, אם votes[num-1] != -1 מגדילים את votes[num-1] ב-1.

ב. מחפשים מינימום רק בין התאים ששונים מ-1-; את התא שנמצא קובעים ל-1- ומקטינים את numOfPart ב-1.

ג. אם numOfPart != 3 מחזירים false; אחרת מדפיסים i+1 לכל תא שערכו שונה מ-1- ומחזירים true.

שאלה 8

א. הפלט: Your seats in 1, 1, 2, 1, 3, 1, 4, 2, 5, 2, 5, 3 (12 שורות; אף פעם לא No seats left). בסוף t.carr = {19, 18, 18, 15, 18}.

ב. עוברים על הקרונות לפי הסדר; מחזירים i+1 עבור הקרון הראשון שבו carr[i] + numPass <= maxInCarr; אם אין כזה מחזירים 1-.

שאלה 9

א. מחרוזת ריקה – false; אחרת עוברים על התווים: תו שאינו בין 'a' ל-'z' או אות שכבר סומנה במערך seen בגודל 26 – false; בסוף true.

ב. $O(n)$, כאשר n הוא אורך המחרוזת (מעבר אחד, עבודה קבועה לכל תו).

שאלה 10

א. למשל: שורה 0: 0, 1, 2, 3, 4; שורה 1: 1, 2, 10, 11, 12; שורה 2: 2, 3, 11, 12, 13.

ב. $0 \le mat[0][0] \le 7$.

ג. כל ערך שלם – התא mat[2][4] אינו נבדק כלל.

ד. בודקת שכל תא שאינו בשורה האחרונה או בעמודה האחרונה אי-שלילי וקטן ממש מהתא שמימינו ומהתא שמתחתיו – כלומר שהמערך ממוין בסדר עולה ממש בשורות ובעמודות (חוץ מהשורה האחרונה והעמודה האחרונה, שלא נבדקות בתוכן עצמן, והפינה הימנית-התחתונה שאינה נבדקת כלל).

שאלה 11

א. יוצרים מערך res בגודל a.length·k, ולכל i ולכל j בין 0 ל-k-1 מציבים res[i·k + j] = a[i].

ב. אם אורך brr אינו כפולה של אורך arr מחזירים 0; אחרת k = brr.length / arr.length, ובודקים לכל i ש-brr[i] == arr[i / k]; אם הכול מתאים מחזירים k, אחרת 0.

ג. $O(m)$, כאשר m הוא אורך brr (כלומר $O(n \cdot k)$).

שאלה 12

א. (1) בנאי עם שישה פרמטרים: מציב name, price, hasMilk ויוצר lastDate = new Date(day,` month, year).`

(2) `return d.before(lastDate) || d.equals(lastDate);`

(3) מחזירה true כאשר `hasMilk == other.hasMilk`, `lastDate.equals(other.lastDate)` ו-`Math.abs(price - other.price) <= 10`.

ב. (1) אם numOfCakes == 0 מחזירים null; אחרת חיפוש מקסימום לפי תאריך תפוגה: best מתחלפת ב-arr[i] כאשר arr[i].getLastDate().before(best.getLastDate()).

(2) עוברים עם i על העוגות; עוגה שתאריך התפוגה שלה before(d) מוחלפת בעוגה האחרונה, התא האחרון מקבל null ו-numOfCakes קטן ב-1 (בלי לקדם את i); אחרת i++.

(3) מעבר ראשון: לכל שם מחפשים עוגה במקרר; שם שלא נמצא מודפס וההזמנה נכשלת – מחזירים 0. אחרת מעבר שני: מדפיסים כל עוגה (toString) וסוכמים את המחירים; מחזירים את הסכום.

שאלה 13

א. הפעולה מחזירה 1.

ב. הפעולה מחזירה "CCAVINE".

ג. הפעולה מחזירה "VNIECCA".

ד. ממיינת את תווי המחרוזת בסדר יורד (מהגדול לקטן) ומחזירה את המחרוזת הממוינת.

שאלה 14

א. עוברים על כל פינה שמאלית-עליונה (i, j) עם i < rows-1 ו-j < cols-1; אם ארבעת התאים [i][j], [i][j+1], [i+1][j], [i+1][j+1] שווים ל-value מחזירים true; בסוף false.

ב. לכל צבע 0 עד 9 בודקים בעזרת הפעולה מסעיף א אם יש כתם בצבע הזה וסופרים את הצבעים; מחזירים true אם מספרם גדול מ-3.

ג. שתיהן $O(n \cdot m)$ (n שורות, m עמודות): סעיף א עובר פעם אחת על כל הפינות; סעיף ב מפעיל את א 10 פעמים – קבוע.

כל מבחני מה"ט עם פתרונות מלאים: www.guyai.co.il/mahat