GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 2025 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-#C (עמוד 2) מבחן ב-Java (עמוד 13) בכל מבחן 14 שאלות. **חלק א' – 48 נקודות** שאלות 1–7: יש לענות על **ארבע** שאלות בלבד. ערך כל שאלה – 12 נקודות. **חלק ב' – 30 נקודות** שאלות 8–11: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 15 נקודות. **חלק ג' – 22 נקודות** שאלות 12–14: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 11 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 המבחן, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת המבחן). 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 להמשיך בפתרון השאלה. נמקו את בחירתכם. **חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!** **מבחן ב-C# – הנחיות כלליות לנבחנים:** 1. דוגמה להוראה לקליטת מספר שלם: `int x = int.Parse(Console.ReadLine());` `int x = Convert.ToInt32(Console.ReadLine());` 2. דוגמה להוראה לקליטת מספר עשרוני: `double y = double.Parse(Console.ReadLine());` 3. דוגמה להוראה לקליטת מחרוזת: `string str = Console.ReadLine();` 4. הוראות לפלט על המסך: `Console.WriteLine();` 5. יצירת מספר שלם אקראי num בין X ל-Y כולל (X <= num <= Y): `Random rand = new Random();` `int num = rand.Next(X, Y + 1);` 6 הפונקציה `s.Substring(k)` מחזירה תת-מחרוזת ממקום k עד סוף המחרוזת. הפונקציה `s.Substring(k,p)` מחזירה תת-מחרוזת באורך p ממקום k (כולל) **מבחן ב-JAVA – הנחיות כלליות לנבחנים:** 7. בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה: `Scanner in=new Scanner (System.in);` 8. דוגמה להוראה לקליטת מספר שלם: `int x = in.nextInt();` 9 דוגמה להוראה לקליטת מספר עשרוני: `double y =` `in.nextDouble();` 10. דוגמה להוראה לקליטת מחרוזת: `String str = in.next();` 11 הוראות לפלט על המסך: `System.out.print();` 12. 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);` 13. הפונקציה `s.substring(k)` מחזירה תת-מחרוזת ממקום k עד סוף המחרוזת. הפונקציה `s.substring(k,p)` מחזירה תת-מחרוזת ממקום k (כולל) עד מקום p (לא כולל).

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 שממשיכה כל עוד sum <= 100 וגם count <= 15; בכל סיבוב קולטים מספר, מוסיפים לסכום, מגדילים את המונה וסופרים חיוביים; בסוף מדפיסים sum/count (כממשי) ואת מספר החיוביים.

שאלה 2

קולטים את המחרוזת הראשונה כארוכה וכקצרה; עבור כל אחת מ-29 הבאות מעדכנים longest/shortest לפי האורך; מונים מחרוזת אם התו הראשון או האחרון בין 'A' ל-'Z'; מדפיסים את שלושת הנתונים.

שאלה 3

60 פעמים מגרילים מספר בין 100 ל-999; מונים אותו כזוגי אם num % 2 == 0, ומונים אותו במונה השני אם פעולת עזר מוצאת בו ספרה זוגית (פירוק בעזרת 10% ו-/10); מדפיסים את שני המונים.

שאלה 4

א. הכלב מתאים אם this.gender == gender, הגיל ידוע (age != 0) ו-this.age <= age.

ב. עוברים על המערך ומדפיסים שם וסוג של כל כלב שעבורו isItSuitable('M', 2) מחזירה true וגם getIsImunned() מחזירה true.

ג. מונה את הכלבים שעבורם getIsImunned() == false וגם getAge() == 0, ומחזירה את המונה.

שאלה 5

א. אם האורך זוגי – false; אחרת mid = length/2, מסכמים את האיברים שלפני mid ואת האיברים שאחריו, ומחזירים true אם שני הסכומים שווים ל-arr[mid].

ב. צד שמאל: half=size/2 מספרים אקראיים בין 100 ל-half/999; המרכז = סכומם; צד ימין: מחלקים את הסכום באקראי ל-half מספרים תלת-ספרתיים (כל אחד בין 100 לערך שמשאיר לפחות 100 לכל הבאים), והאחרון משלים. אפשרי רק כאשר size <= 19.

שאלה 6

א. סכום הערכים בתאים 91 עד 100.

ב. עוברים מהתא 100 כלפי מטה ומחזירים את האינדקס הראשון שערכו גדול מ-0 (או -1 אם המערך ריק מציונים).

ג. חיפוש מקסימום של הערכים, מהאינדקס 100 כלפי מטה, עם השוואה חזקה (<) – כך בשוויון נשאר הציון הגבוה; מחזירים את האינדקס.

שאלה 7

א. "K%ABC?D"

ב. כן, למשל "AB" (וגם "A").

ג. כן, למשל str1="AA" ו-str2="A": שתיהן מחזירות "A".

ד. מצמצמת כל רצף של אות גדולה זהה לאות אחת, ומכל רצף של תו אחר (שאינו אות גדולה) משמיטה את התו האחרון ברצף (תו בודד נמחק); התו האחרון של המחרוזת נשאר תמיד.

שאלה 8

א. לכל איבר סופרים כמה איברים קטנים ממנו; האיבר שעבורו המספר שווה ל-length/2 הוא החציון. סיבוכיות $O(n^2)$.

ב. מוצאים את החציון med (סעיף א) ושמים אותו ב-res[n/2]; עוברים על arr: איבר קטן מ-med נכתב ב-res[left++] (מ-0), איבר גדול נכתב ב-res[right++] (מ-n/2+1).

ג. median: $O(n^2)$; buildSpecial: $O(n^2)$ (קריאה ל-median ועוד מעבר אחד $O(n)$).

שאלה 9

א. בדיקות לפי הסדר: משקל וגובה שווים – 0; שניהם גדולים – 1; רק המשקל גדול – 2; רק הגובה גדול – 3; אחרת 4.

ב. lastName שווה, firstName שונה, id שונה, ו-birth שווה או birth.tomorrow() שווה ל-other.birth או other.birth.tomorrow() שווה ל-birth.

ג. מדפיסים את getId() של כל babies[i] שעבורו babies[i].compareTo(baby) == 0.

ד. לולאה כפולה על כל הזוגות i<j, וסופרים את הזוגות שעבורם babies[i].areTwins(babies[j]) מחזירה true.

שאלה 10

א. (1) למשל 1234.

(2) מחזירה true אם ספרות המספר עולות ממש משמאל לימין (כל ספרה גדולה מזו שמשמאלה); אחרת false. מספר חד-ספרתי – true.

ב. (1) 5

(2) מחזירה את המקום (האינדקס) של האיבר הגדול ביותר מבין האיברים שספרותיהם עולות ממש; אם אין כזה – -1.

ג. (1) למשל {12, 3, 45, 789, 6, 123}.

(2) אם כל איברי המערך בעלי ספרות עולות ממש – מחזירה את מקום האיבר הגדול ביותר במערך; אחרת מחזירה -1.

שאלה 11

א. return red == 0 && green == 0 && blue == 0;

ב. סופרים בלולאה כפולה פיקסלים לבנים (isWhite) ושחורים (isBlack) ומחזירים white == black.

ג. לולאה כפולה; אם נמצא פיקסל שאינו שחור וגם אינו לבן – false מיד; בסוף – true.

ד. לולאה k=0..255: צובעים בלבן (red=green=blue=255 בעזרת set) את pixels[0][k], pixels[255][k], pixels[k][0], pixels[k][255].

שאלה 12

א. Clinic: name, phone, status (1 – מבוגרים, 2 – ילדים, 3 – שניהם) ובנאי עם שלושה פרמטרים; Center: name, city, Clinic[15] clinics, numClinics ובנאי (name, city); Chain: Center[40] centers ובנאי.

ב. פעולת עזר findClinic מחפשת את המרפאה לפי שם (או null); treatsChildren מחזירה false אם לא נמצאה, ואחרת status != ADULTS.

ג. לכל מרכז: findClinic = eyes("עיניים"); אם addClinic – null(new Clinic("עיניים", "", BOTH)) (נכשלת בשקט אם יש 15 מרפאות); אם status == ADULTS – setStatus(BOTH); אחרת לא משנים.

שאלה 13

א. -1

ב. משווה את המחרוזות לפי סדר מילוני: -1 אם s1 קודמת ל-s2, 0 אם הן שוות, 1 אם s1 באה אחרי s2.

ג. {"WORLD", "TEST", "FOR", "DAY", "GOOD", "HELLO"}

ד. מחזירה מערך חדש: בתחילתו המחרוזות הקודמות ל-s בסדר מילוני (בסדר הופעתן), ובסופו המחרוזות שאינן קודמות ל-s (שוות או באות אחריו) בסדר הפוך להופעתן.

שאלה 14

א. בודקים שכל שורה באורך n (ריבועי); לכל mat[i][j] בודקים שהוא חיובי ושהספרה j מופיעה בו (פעולת עזר hasDigit בעזרת 10% ו-/10); בכישלון – false, בסוף – true.

ב. $O(n^2)$ עבור מערך n×n (בכל אחד מ-n^2 האיברים בדיקת ספרות באורך מספר הספרות d, כלומר $O(n^2 \cdot d)$, ו-d חסום – לכל היותר 10 ספרות ל-int).

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