GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 2021 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. עליך לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:** מבחן ב- Java (עמוד 2) מבחן ב- C# (עמוד 15) בכל מבחן 13 שאלות. **חלק א' - 40 נקודות** שאלות 1-6: יש לענות על **ארבע** שאלות בלבד. ערך כל שאלה - 10 נקודות. **חלק ב' - 24 נקודות** שאלות 7-9: יש לענות על **שתיים** שאלות בלבד. ערך כל שאלה - 12 נקודות. **חלק ג' - 36 נקודות** שאלות 10-13: יש לענות על **שתיים** 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 לדעתך חסר בשאלה נתון, יש לציין זאת ולהוסיף נתון מתאים שיאפשר לך להמשיך בפתרון השאלה. נמק את בחירתך. **הנחיות כלליות לנבחנים - מבחן ב- JAVA:** 1. בכל שאלה שנדרשת בה קליטה, אין צורך בבדיקת תקינות הקלט. 2. בכל שאלה שנדרשת בה קליטה, הנח שבתוכנית כתובה ההוראה: `Scanner in=new Scanner(System.in);` 3. דוגמה להוראה לקליטת מספר שלם: `int x = in.nextInt();` 4. דוגמה להוראה לקליטת מספר עשרוני: `double y = in.nextDouble();` 5. דוגמה להוראה לקליטת מחרוזת: `String str = in.next();` 6. הוראות לפלט על המסך: `System.out.print` ו-`System.out.println();` **הנחיות כלליות לנבחנים - מבחן ב- C#:** 1. בכל שאלה שנדרשת בה קליטה, אין צורך בבדיקת תקינות הקלט. 2. דוגמה להוראה לקליטת מספר שלם: `int x = int.Parse(Console.ReadLine());` 3. דוגמה להוראה לקליטת מספר עשרוני: `double y = double.Parse(Console.ReadLine());` 4. דוגמה להוראה לקליטת מחרוזת: `String str = Console.ReadLine();` 5. הוראות לפלט על המסך: `Console.Write` ו-`Console.WriteLine();`

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

בהצלחה!

תשובות סופיות

שאלה 1

א. בתוך לולאה של 40 קליטות: אם $100 \le num \le 999$ מדפיסים `num%10 + num/10%10 + num/100`.

ב. מונה countEven (גדל כאשר `num % 2 == 0`) וצובר sumOdd (מוסיפים לו את num אחרת), מאופסים לפני הלולאה ומודפסים אחריה.

שאלה 2

א. בנאי שמציב `this.color = color` ו-10 בשלוש התכונות length, width, height.

ב. מעבר על המערך, חישוב נפח `getHeight()*getWidth()*getLength()` לכל קופסה ושמירת המקום של הנפח הקטן ביותר; מחזירים את `getColor()` של הקופסה במקום זה.

שאלה 3

א. `public Tank(int capacity)` שמציב את הקיבולת ו-`amount = 0`.

ב. `return this.amount > 0;`

ג. '+': אפשרי אם $amount + num \le capacity$ (מוסיפים ומחזירים true); '-': אפשרי אם $num \le amount$ (מורידים ומחזירים true); אחרת false בלי שינוי.

ד. המינימום בין הדלק שיש ב-t1 לבין המקום הפנוי ב-t2:
$\min(t1.amount,\ t2.capacity - t2.amount)$.

שאלה 4

א. בודקים: אורך אי-זוגי; `arr[n/2]==0`; כל איבר לפני האמצע בין 1 ל-9; אף איבר אחרי האמצע אינו בין 1 ל-9. כשל באחד – false.

ב. $O(n)$ – כל איבר נבדק לכל היותר פעם אחת (שתי לולאות על חצאי המערך).

שאלה 5

א. (1) false

(2) למשל num = 9 (כי $2 + 7 = 9$) – הפעולה תחזיר true.

ב. מחזירה true אם יש במערך שני איברים (בשני מקומות שונים) שסכומם שווה ל-num, אחרת false.

ג. $O(n)$ – בכל סיבוב k גדל או p קטן, ולכן יש לכל היותר $n - 1$ סיבובים.

שאלה 6

false אם התו הראשון או האחרון הוא נקודה (או מחרוזת ריקה); מעבר על התווים הפנימיים: נקודה שאחריה נקודה – false; בסוף מחזירים אם נמצאה נקודה לפחות פעם אחת.

שאלה 7

א. (1) בנאי עם ארבעה פרמטרים שמציב כל פרמטר בתכונה המתאימה.

(2) בנאי `Battery(double volt, char size)` שמציב את המתח והגודל, `manufacturer = "Duracell"` ו-`isNormal = true`.

ב. (1) משווים את sum ל-`pack.getPrice()`: גדול - "אפשר לרכוש" ו-sum–price; קטן - "אי אפשר לרכוש" ו-price–sum; שווה - "O.K.".

(2) פעולה פנימית ב-BatteryPack: צוברים את `getVolt()` של כל סוללה שאינה null ו-`getIsNormal()` אמת; מחזירים `total >= v`.

שאלה 8

א. נוצרים 4 עצמים (d[4] מפנה לאותו עצם כמו d[3]); tests של Avi: 7→4, של Dani: 2→7. הפלט: 3, 4, 1, 7, 7, 4.4, 3 (כל אחד בשורה).

ב. מעבר ראשון סופר כמה נהגים עם `getTests() < num`; יוצרים מערך String בגודל זה; מעבר שני מעתיק את שמותיהם (`getName()`) ומחזירים אותו.

שאלה 9

א. בונים מחרוזת של האותיות A..Z בלבד (לפי הסדר) ובודקים בשני מצביעים שהיא פלינדרום; אם אין אותיות גדולות - false.

ב. מעבר על המערך; לכל מחרוזת שהפעולה מסעיף א מחזירה עבורה true משווים את אורכה לאורך הארוכה שנמצאה עד כה ומעדכנים; מחזירים אותה.

שאלה 10

א. פירוק המספר (בערך מוחלט) לספרות בלולאת do-while וספירת הספרות הזוגיות; עבור 0 מוחזר 1.

ב. מקסימום של `parityDegree(arr[i])` על כל איברי המערך.

ג. מחשבים sup = דרגת העל; מערך מונים `count` בגודל sup+1 סופר כמה איברים יש מכל דרגה; מחזירים true רק אם `count[d] == 1` לכל $0 \le d \le sup$.

ד. א: $O(k)$ (k = מספר הספרות, לכל היותר 10 ב-int, כלומר $O(1)$); ב: $O(n \cdot k) = O(n)$; ג: $O(n \cdot k + sup) = O(n)$.

שאלה 11

א. (1) b = {84, 74, 69, 66, 50, 30}

(2) כל תא a[i] בתחום start..stop הופך לסכום האיברים המקוריים מ-a[i] ועד a[stop] (סכום "סיפא"); a[stop] ושאר התאים לא משתנים.

ב. (1) c = {−1, 11, −3, 15, 5, 10}

(2) כל תא a[i] עבור start ≤ i < stop הופך לסכום שלו ושל השכן הימני המקורי a[i+1]; a[stop] לא משתנה.

שאלה 12

א. אם $num \geq minutes$: מחזירים $num - minutes$, ומציבים minutes = 0 ו-isComplete = true; אחרת num -= minutes ומחזירים 0.

ב. מחפשים את התא הראשון שהוא null - מכניסים, true ,numOfTasks++; אם אין, מחפשים משימה עם isComplete - מחליפים (numOfTasks לא משתנה), true; אחרת false.

ג. חיפוש מקסימום של priority בין התאים שאינם null ו-isComplete שלהם false; index מאותחל ל-1- ומוחזר.

ד. לולאה כל עוד i = importantTask : num > 0(); אם i == 1- מסיימים; אחרת num = tasks[i].doIt(num).

שאלה 13

א. false

ב. למשל "olleh" = "s3" - יוחזר true.

ג. מחזירה true אם s2 מורכבת בדיוק מאותם תווים כמו s1 (כל תו באותו מספר פעמים, בסדר כלשהו) - כלומר s2 היא תמורה (אנגרמה) של s1; אחרת false.

ד. בודקים שהאורכים שווים; בלולאה על התווים של s1: מחפשים כל תו ב-(indexOf) s2, אם אינו נמצא - false, אחרת מוחקים אותו מ-s2; בסוף true.

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