GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2024 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 14) 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 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 להמשיך בפתרון השאלה. נמקו את בחירתכם. **חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!** **מבחן ב- JAVA – הנחיות כלליות לנבחנים: 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. 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).**

מבחן ב- C# – הנחיות כלליות לנבחנים: 1. דוגמה להוראה לקליטת מספר שלם: `int x = Convert.ToInt32(Console.ReadLine());` `int x = int.Parse(Console.ReadLine());` 2. דוגמה להוראה לקליטת מספר עשרוני: `double y = double.Parse(Console.ReadLine());` 3. דוגמה 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). **בשאלון זה 25 עמודים ו- 2 עמודי מילון.

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

בהצלחה!

תשובות סופיות

שאלה 1

לולאה של 20 סיבובים, בכל סיבוב קולטים שני מספרים; מונה למספרים בין 10 ל-99 וצובר למספרים זוגיים, ובסוף מדפיסים את שניהם.

שאלה 2

לולאת while שרצה כל עוד אורך המחרוזת זוגי: לכל מחרוזת בודקים indexOf('Z') ומדפיסים הודעה, ומונים מחרוזות שבהן התו הראשון שווה לאחרון; בסוף מדפיסים את המונה.

שאלה 3

מערך מונים count בגודל 100 (count[x] = מספר הפעמים שנוצר x); מגרילים 40 מספרים בין 10 ל-99 ומגדילים את המונה; מחפשים את המיקום עם המונה המרבי ומדפיסים את כל x שהמונה שלו 0.

שאלה 4

א. מחזירים true אם ורק אם this.isMilk == other.isMilk, this.raiting == other.raiting וגם other.price <= this.price.

ב. עוברים על המערך ומדפיסים את arr[i].getName() לכל i שעבורו arr[i].canChange(newFood) מחזירה true.

ג. לולאה כפולה על כל הזוגות i < j; אם arr[i].getPrice() + arr[j].getPrice() < 100 מדפיסים את שני השמות.

ד. סעיף ב: $O(n)$; סעיף ג: $O(n^2)$ (n – מספר המנות במערך).

שאלה 5

א. אם k > arr.length מחזירים false; אחרת סוכמים את arr[0] עד arr[k-1] ומחזירים sum % k == 0.

ב. לולאה על k מ-1 עד arr.length; אם isPerfectK(arr, k) מחזירה false עבור k כלשהו – מחזירים false, ואחרי הלולאה מחזירים true.

ג. סעיף א: $O(k)$, ובמקרה הגרוע $O(n)$; סעיף ב: $O(n^2)$ (n – אורך המערך).

שאלה 6

א. (1) למשל num = 7707.

(2) מחזירה כמה פעמים הספרה dig מופיעה במספר num.

ב. (1) הפעולה מחזירה 6.

(2) מחזירה כמה פעמים הספרה dig מופיעה בסך הכול בכל איברי המערך arr.

ג. (1) למשל arr = {6, 6, 6, 6, 6, 6}.

(2) מחזירה את הספרה (0-9) שמופיעה הכי הרבה פעמים בכל איברי המערך; בתיקו – הספרה הקטנה מביניהן.

שאלה 7

א. למשל "123a3A"; הפעולה מחזירה 4.

ב. למשל "123456"; הפעולה מחזירה 6.

ג. למשל "Aa12Aa"; יודפס 1#0%?5#4?%.

ד. מדפיסה בין שני סימני % את המקום של כל אות באנגלית (עם # לאות גדולה ו-? לאות קטנה), ומחזירה את מספר התווים שאינם אותיות באנגלית.

שאלה 8

א. שני מצביעים i מההתחלה ו-j מהסוף; כל עוד i < j משווים arr[i] ל-arr[j] – אם שונים מחזירים false, אחרת מקדמים את שניהם; בסוף מחזירים true.

ב. שני מצביעים כמו בסעיף א, אבל מדלגים על איברים אי-זוגיים: אם arr[i] אי-זוגי מקדמים i, אם arr[j] אי-זוגי מורידים j, ואם שניהם זוגיים – משווים (שונים → false) ומקדמים את שניהם; בסוף true.

ג. שתיהן $O(n)$ (n – אורך המערך).

שאלה 9

א. שתי פעולות בונות – Destination(String name) ו-Destination(String name, double price, int num); getName, getPrice, getNum; setName, setPrice, setNum.

ב. יודפס 60000.0 ואחריו 5000.0 (ב-C#: 60000 ו-5000).

ג. חיפוש מינימום על getNum(): שומרים את מיקום היעד עם הכי מעט כרטיסים ומחזירים את שמו.

ד. מוסיפים לכותרת פרמטר double maxPrice; מעבר ראשון סופר כמה יעדים זולים מ-maxPrice, יוצרים מערך מחרוזות בגודל הזה (אולי 0), ובמעבר שני ממלאים אותו בשמות.

שאלה 10

א. arr = {5, 7, 40, 26, 21, 15, 12, 51, 71, 84}.

ב. הופכת את סדר האיברים במערך arr בקטע שבין המקום start למקום end (כולל); שאר המערך לא משתנה.

ג. arr = {40, 51, 71, 84, 5, 7, 12, 15, 21, 26}.

ד. כן – כל s שהוא כפולה של אורך המערך (10, 20, 30, ...).

ה. מזיזה את איברי המערך arr באופן מעגלי ימינה ב-s מקומות (כלומר ב-arr.length % s מקומות).

שאלה 11

א. (1) למשל mystery(24, 15).

(2) למשל mystery(600, 22).

(3) למשל mystery(12, 234).

(4) לא. לכל x תלת-ספרתי, mystery(x, -x) מחזירה 2.

(5) לא. לכל x (שונה מ-0), mystery(x, 0) מחזירה 3.

ב. pos1 = 5, pos2 = 2, ולכן מוחזר false.

ג. מחזירה true אם המספר הראשון עם הכי הרבה ספרות הוא גם המספר הראשון עם סכום הספרות הגדול ביותר (אותו מיקום); אחרת false.

שאלה 12

א. שלוש פעולות: isSuperEvenNum(num) – חיובי, כל הספרות זוגיות ומספר הספרות זוגי; isSuperEvenCol(mat, col) – מספר השורות זוגי ויותר ממחצית איברי העמודה סופר זוגיים; isSuperEvenMat(mat) – מספר העמודות זוגי וכל עמודה סופר זוגית.

ב. $O(n \cdot m)$ (n – מספר השורות, m – מספר העמודות).

שאלה 13

א. Major: int code, String majName; College: String name, String city, Major[] majors (בגודל 10), int numMajors; Network: College[] colleges (80 מכללות).

ב. עוברים על colleges; לכל מכללה ש-getCity() שווה לשם היישוב מדפיסים את getName() ומסמנים דגל found; אם בסוף found הוא false – מדפיסים הודעה שאין מכללה ביישוב.

ג. פעולת עזר hasMajor(code) במחלקה College; ב-Network: מכללה מתאימה אם getNumMajors() > 10 וגם !hasMajor(code). סופרים את המתאימות; אם 0 מחזירים null, אחרת יוצרים מערך בגודל המונה וממלאים בשמות.

שאלה 14

א. (1) brr = {-1, 16, -11, 14, 2, 8}.

(2) כל איבר במקומות begin עד end-1 מוחלף בסכום לסירוגין (פלוס, מינוס, פלוס...) של האיברים המקוריים ממקומו ועד end; האיבר במקום end לא משתנה.

ב. (1) crr = {6, 5, 7, 8, 10, 3}.

(2) מחליפה כל איבר במקומות begin עד end-1 בהפרש בינו לבין האיבר המקורי שאחריו; האיבר במקום end לא משתנה.

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