GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2026 מועד א'
סמל השאלון: 97103
קובץ: תשובות סופיות

אלגוריתמיקה ותכנות (Python)

הנדסת תוכנה · הנדסאים וטכנאים

הוראות

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: מבחן ב-Python (עמוד 2) בכל מבחן 10 שאלות. חלק א' – 36 נקודות שאלות 1–4: יש לענות על **שלוש** שאלות בלבד. ערך כל שאלה – 12 נקודות. חלק ב' – 30 נקודות שאלות 5–7: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 15 נקודות. חלק ג' – 34 נקודות שאלות 8–10: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה – 17 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. הוראות מיוחדות: 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. הנחיות כלליות לנבחנים ב-Python: 1. דוגמה להוראה לקליטת מספר שלם: `x = int(input())` 2. דוגמה להוראה לקליטת מספר עשרוני: `y = float(input())` 3. דוגמה להוראה לקליטת מחרוזת: `str = input()` 4. יצירת מספר שלם אקראי num בין X ל-Y כולל `(X <= num <= Y)`: `num = random.randint(x,y)` ... `import random`

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. משווים כל זוג תווים במקומות $i < j$; אם נמצא זוג שווה מחזירים False, ואם לא נמצא – True.

ב. לולאת while עם קליטה לפני הלולאה ובסופה; לכל מחרוזת שאינה מחרוזת הסיום מזמנים את `is_unique` ומגדילים מונה; בסוף מדפיסים את המונה.

שאלה 2

א. כל עוד המספר גדול או שווה ל-10 מחלקים אותו ב-10 בחלוקת שלמים (`//`); מה שנשאר הוא הספרה המובילה.

ב. עוברים על כל מקום התחלה i מ-0 עד `len(lst) - 3`, ומונים את המקומות שבהם `leading_digit` של `lst[i+2]`, `lst[i+1]`, `lst[i]` שווה.

שאלה 3

א. (1) count = 5, ו-`len(arr) - 1 = 6`, ולכן הפונקציה מחזירה False.

(2) כן. למשל `[1, -5, 2, -8, 4, -3, 10]` – הסימנים מתחלפים בכל זוג סמוך, ו-secret מחזירה True.

ב. מחזירה True אם הסימנים ברשימה מתחלפים – לכל שני איברים סמוכים סימנים מנוגדים (בלי אפסים); אחרת False.

ג. ברשימה עם 2 איברים לפחות הטענה נכונה (כל זוג שמכיל 0 נותן מכפלה 0, שאינה נספרת). אבל באופן כללי היא אינה נכונה: עבור `[0]` מוחזר True.

שאלה 4

א. צובר: עוברים על הטיסות, ולכל טיסה עם `f.destination == dest` מוסיפים את `f.passengers`; מחזירים את הצובר.

ב. לכל טיסה מחשבים מקומות פנויים `capacity - passengers`; אם היעד הוא dest והמקומות הפנויים $\geq num$ – מדפיסים את `flight_Num`.

שאלה 5

א. עוברים על המקומות $j < k$; אם `arr[j] >= arr[k]` מחזירים False; אם הלולאה הסתיימה – True.

ב. יוצרים רשימה ריקה; לכל k מ-0 עד הסוף, אם `is_breakthrough(arr, k)` – מוסיפים את `arr[k]` לסוף; מחזירים את הרשימה.

ג. (1) 1 (עבור $n \geq 1$). דוגמה: `[9, 3, 5, 1]` → `[9]`.

(2) n. דוגמה: `[1, 2, 3, 4]` → `[1, 2, 3, 4]`.

שאלה 6

א. `__init__(self, donor_name, amount, donation_type)`: נשמר אם $amount > 0$, אחרת `amount` = 0; `donation_type` נשמר אם הוא אחד מ-'M', 'C', 'F', אחרת 'M'.

ב. צובר על `self.donations`: לכל תרומה `d` עם `d.donation_type == d_type` מוסיפים את `d.amount`; מחזירים את הצובר (0 אם אין תרומות מהסוג).

ג. חיפוש מקסימום: לכל פרויקט מחשבים `p.get_type_total(d_type)`, שומרים את הפרויקט עם הסכום הגדול ביותר, ומדפיסים את ה-location שלו.

שאלה 7

א. (1) הפונקציה מחזירה 5 (החפיפה "storm").

(2) למשל `secret("settle", "sunset")` מחזירה 3.

(3) מחזירה את האורך של הסיומת הארוכה ביותר של s1 שהיא גם רישא (התחלה) של s2 – אורך החפיפה; 0 אם אין חפיפה.

ב. (1) `secret("Roommate", "ballroom")` – `False`, מחזירה 0, כי "room" שונה מ-"Room" (R גדולה).

(2) למשל `["antfox", "beeant", "cowbee", "pigcow", "dogpig", "catdog"]` – לכל זוג סמוך חפיפה של 3 בדיוק.

שאלה 8

א. במעבר אחד מחשבים את המקסימום של תאי המסגרת ואת המינימום של התאים הפנימיים; מחזירים `max_frame < min_inner`.

ב. $O(n \cdot m)$ עבור מטריצה בגודל $n \times m$ (עבור מטריצה ריבועית $O(n^2)$).

ג. מחלקים את הטווח בנקודה `mid = (low + high) // 2`: תאי המסגרת מקבלים `random.randint(low, mid)` והתאים הפנימיים `random.randint(mid + 1, high)`.

שאלה 9

א. (1) `__init__(self, course_name, min_entry_grade, max_students)` שומרת את שלושת הערכים ומאתחלת `self.registered = []`.

(2) אם `len(self.registered) < self.max_students` וגם `grade >= self.min_entry_grade` – מוסיפים `Student(name, grade)` ל-`registered` ומחזירים True; אחרת False.

(3) אם אין רשומים – False; אחרת סופרים את הרשומים עם $grade > 90$ ומחזירים `count >= 2`.

ב. (1) עוברים על `self.courses` ומדפיסים `c.course_name` לכל קורס שבו `len(c.registered) >= c.max_students`.

(2) יוצרים רשימה ריקה, ולכל קורס `c` שעבורו `c.is_honors_course()` מחזירה True מוסיפים את `c.course_name`; מחזירים את הרשימה.

שאלה 10

א. (1) True.

(2) False.

(3) עבור $num < 10$ (חד-ספרתי, 0 או שלילי) – False. עבור $num \geq 10$ – True אם הסכום לסירוגין של הספרות הוא 0, כלומר סכום הספרות במקומות האי-זוגיים (אחדות, מאות, ...) שווה לסכום הספרות במקומות הזוגיים (עשרות, אלפים, ...).

ב. (1) 1012.

(2) מחזירה את המספר המיוחד הגדול ביותר ברשימה (מהמקום i ועד הסוף); אם אין מספר מיוחד – מחזירה -1.

ג. `all_special(arr, i=0)`: אם `i == len(arr)` – True; אם `arr[i]` אינו מיוחד – False; אחרת מחזירים `all_special(arr, i + 1)`.

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