GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 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: בשאלון זה שני מבחנים. עליכם לענות על מבחן אחד בלבד בהתאם למוסד הלימודים: 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. מילון עזר.

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: דוגמה להוראה לקליטת מספר שלם: `x = int(input());` דוגמה להוראה לקליטת מספר עשרוני: `y = float(input());`. דוגמה להוראה לקליטת מחרוזת: `str = input();` יצירת מספר שלם אקראי num בין X ל-Y כולל (`X <= num <= Y`): ```code {"text": "import random\n...\nnum = random.randint(x,y)"}```

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. עוברים על התווים: אות גדולה ('A'–'Z') מגדילה מונה אחד, אות קטנה ('a'–'z') מונה שני, וכל תו אחר מחזיר מיד `False`; בסוף מחזירים `lower > upper`.

ב. לולאה שקולטת מחרוזת, מגדילה מונה אם `s[0]` == 'A' או `s[-1]` == 'A', ונעצרת כש-`is_valid(s)` מחזירה `True`; אחרי הלולאה מדפיסים את המונה.

שאלה 2

א. מספר שאינו חיובי – `False`; אחרת מפרקים את המספר לספרות מימין לשמאל ומשווים את הזוגיות של כל ספרה לזו של הספרה שלפניה; זוגיות זהה – `False`, אחרת בסוף `True`.

ב. עוברים על הרשימה ומוסיפים לסכום כל מספר ש-`is_alternating` מחזירה עבורו `True`; מחזירים את הסכום.

שאלה 3

א. (1) הפונקציה מחזירה `False`.

(2) למשל `num = 13` (כי $9 + 4 = 13$); הפונקציה מחזירה `True`.

ב. מחזירה `True` אם יש ברשימה שני איברים (במקומות שונים) שסכומם שווה ל-`num`, ואחרת `False`.

ג. כן, ההנחה הכרחית (למשל `secret([2, 4, 9], 13)` מחזירה `False` אף ש- $4 + 9 = 13$). הפונקציה החדשה מיינת עותק של הרשימה בסדר יורד ומריצה עליו את אותו אלגוריתם.

שאלה 4

א. `__init__(self, truck_id, driver_name)` שומרת את שני הפרמטרים, וקובעת `max_weight = 4000`, `is_refrigeration = False`, `is_free = True`.

ב. לולאה על הרשימה; אם `t.is_refrigeration` וגם `t.max_weight >= 10000` – מדפיסים `t.truck_id`.

ג. חיפוש מקסימום לפי `max_weight` רק בין המשאיות עם `is_free` ו-`is_refrigeration`; מחזירים את `driver_name` של המשאית שנמצאה, או "NONAME" אם לא נמצאה אף אחת.

שאלה 5

א. סוכמים את כל `d` בתחום 1 עד `num // 2` שמקיימים `num % d == 0`; עבור 12, 6, 10 מתקבלים 8, 6, 16.

ב. `s = sum_of_proper_divisors(num)`; מחזירים 'P' אם `num == s`, 'D' אם `num > s`, ואחרת 'A'.

ג. עוברים על `num` מ-1 עד `limit - 1` ומוסיפים לרשימה כל מספר ש-`check_number_type(num) == num_type`; עבור 60 ו-'A' מתקבלת הרשימה `[12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56]`.

שאלה 6

א. אם האורכים שונים – `False`; אחרת `k = k % len(s1)` ומחזירים `s1[k:] + s1[:k] == s2`.

ב. מנסים `k` מ-1 עד `len(s1)` בעזרת `is_k_rolling`; מחזירים את ה-`k` הראשון שמתאים, ואם אין – `-1`.

ג. לכל מחרוזת `word` ב-`arr`: אם `find_rolling_k(s, word) == -1` מחזירים `False`; אם כולן עברו – `True`.

שאלה 7

א. `out1 = [71, 23, 15, 25, 9]`. אחרי שלב 1: `[0, 1, 0, 1, 0, 2, 0, 0, 0, 1]`; אחרי שלב 2: `[0, 1, 1, 2, 2, 4, 4, 4, 4, 5]`; אחרי שלב 3: `[0, 0, 1, 1, 2, 2, 4, 4, 4, 4]`.

ב. `out2 = [9, 15, 23, 25, 71]`.

ג. מחזירה רשימה חדשה עם איברי `arr` ממוינים בסדר עולה לפי הספרה במקום `exp` (אחדות, עשרות, ...), מיון יציב (מיון מנייה – שלב אחד של מיון בסיס).

שאלה 8

א. אם האיבר בשורה או בעמודה שבשולי המטריצה – `False` (אין לו ארבעה עלי כותרת); אחרת מחזירים אם `arr[row][col]` שווה לסכום ארבעת האיברים האלכסוניים `arr[row±1][col±1]`.

ב. שתי לולאות מקוננות על כל המיקומים הפנימיים (`row` מ-1 עד `len(arr) - 2`, `col` מ-1 עד `len(arr[row]) - 2`); כל איבר ש-`is_flower` מחזירה עבורו `True` נוסף לרשימה; בדוגמה מוחזר `[6, 15]`.

ג. `is_flower`: $O(1)$; `flower_hearts`: $O(n \cdot m)$ עבור מטריצה בת n שורות ו-m עמודות.

שאלה 9

א. (1) מחלקה `Parking` עם שלוש תכונות: `cars` (רשימת עצמי `CompanyCar`), `parking_spots` (רשימת מחרוזות מקבילה – `parking_spots[i]` הוא המקום של `cars[i]`) ו-`limits` (רשימת עצמי `CompanyLimit`).

(2) `__init__(self, limits)`: `self.cars = []`, `self.parking_spots = []`, `self.limits = limits`, ``self.capacity = 150`.

ב. `return spot not in self.parking_spots` – המקום פנוי אם הוא לא מופיע ברשימת המקומות התפוסים.

ג. בודקים `check_spot(spot)` (תפוס – הודעה ו-`False`); מוצאים ב-`limits` את המכסה של `company` וסופרים ב-`cars` את רכבי החברה (מכסה נוצלה – הודעה ו-`False`); אחרת יוצרים `CompanyCar`, מוסיפים אותו ל-`cars` ואת `spot` ל-`parking_spots`, ומחזירים `True`.

שאלה 10

א. (1) `False`.

(2) `b = 27` (או `b = 81`).

(3) מחזירה `True` אם `b` הוא חזקה שלמה אי-שלילית של `a` ($b = a^k$, $k \geq 0$), ואחרת `False`.

ב. (1) `False`.

(2) `n = 16` (או כל ריבוע דו-ספרתי: 25, 36, 49, 64, 81).

(3) עבור $n \geq 0$ מחזירה `True` אם `n` ריבוע שלם (0, 1, 4, 9, ...) ו-`False` אחרת; עבור $n < 0$ הרקורסיה אינה נעצרת (`RecursionError`).

ג. (1) `121`

(2) `1213121`

(3) `121312141213121`

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