GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: אביב 2026 מועד א'
סמל השאלון: 97103
שם התלמיד/ה: _____

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: 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`): ```` ```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!

השאלות

1. "מחרוזת תקינה" היא מחרוזת הכוללת **רק** אותיות (גדולות וקטנות) ומספר האותיות הגדולות שבה **גדול** ממספר האותיות הקטנות.

א. כתבו פונקציה המקבלת מחרוזת ובודקת אם היא "תקינה". אם כן - הפונקציה תחזיר ערך `True`, ואם לא - הפונקציה תחזיר ערך `False`.

(6 נקודות)

ב. כתבו קטע תוכנית הקולט מחרוזות. הקלט מסתיים כאשר תיקלט מחרוזת "תקינה". הקטע יחשב וידפיס את מספר המחרוזות אשר מתחילות **או** מסתיימות באות 'A'.

(6 נקודות)

2. "מספר מתחלף" הוא מספר **שלם חיובי** שבו כל זוג ספרות שכנות (צמודות) הוא בעל זוגיות שונה. כלומר אם ספרה אחת זוגית, הספרה הסמוכה לה תהיה אי-זוגית, ולהפך. מספר חד ספרתי הוא "מספר מתחלף".

לדוגמה:

- המספר **18345** הוא מספר מתחלף.
- המספר **2347** הוא מספר מתחלף.
- המספר **4** הוא מספר מתחלף.
- המספר **23441** אינו מספר מתחלף.
- המספר -1234 אינו מספר מתחלף.

א. כתבו פונקציה המקבלת מספר שלם ובודקת אם הוא מספר מתחלף. אם כן - הפונקציה תחזיר ערך `True`, ואם לא - הפונקציה תחזיר ערך `False`.

(6 נקודות)

ב. כתבו פונקציה המקבלת רשימה של מספרים שלמים ומחזירה סכום של כל המספרים המתחלפים הנמצאים ברשימה.

(6 נקודות)

3. הגדרה: רשימה נקראת "רשימה ממוינת בסדר יורד" אם כל איבר (פרט לאיבר הראשון) קטן מהאיבר הקודם לו או שווה לו.

נתונה הפונקציה `secret` המקבלת רשימת מספרים שלמים ממוינת בסדר יורד ומספר שלם נוסף:

```
def secret(arr, num):
    i = 0
    j = len(arr) - 1

    while i < j:
        s = arr[i] + arr[j]
        if s > num:
            i += 1
        else:
            if s < num:
                j -= 1
            else:
                return True

    return False
```

א. נתונה הרשימה `arr=[20, 16, 9, 4, 2]`

(1) עקבו בעזרת טבלת מעקב אחר הזימון `secret(arr, 12)`, ורשמו מה תחזיר הפונקציה.

(2) תנו דוגמה לפרמטר `num` כך שזימון `secret (arr, num)` יחזיר תוצאה שונה מהתוצאה שהתקבלה בסעיף א.1.

שימו לב: אין לשנות את הרשימה `arr`!

(5 נקודות)

ב. מה מבצעת הפונקציה `secret` באופן כללי?

(2 נקודות)

ג. האם ההנחה שרשימה `arr` ממוינת בסדר יורד היא הכרחית לביצוע תקין של הפונקציה `secret`?

אם לא - תסבירו למה דרישה זו מיותרת, ואם כן - כתבו פונקציה **חדשה** המבצעת את אותה המשימה עבור כל רשימה של מספרים שלמים (גם ממוינת וגם לא ממוינת).

(5 נקודות)

4. חברת משאיות המתמחה בהובלת מזון החליטה למחשב את ניהול המשאיות שלה. לשם כך הוגדרה המחלקה `Truck`. למחלקה התכונות הבאות:

- מספר הרישוי - `truck_id` - מחרוזת (`str`)
- שם הנהג `driver_name` - מחרוזת (`str`)
- המשקל המרבי (מקסימלי) לאחסון - `max_weight` - מספר שלם (`int`)
- האם המשאית מיועדת לקירור? `is_refrigeration` - בוליאני (`bool`)
- האם המשאית פנויה להובלה? - `is_free` - בוליאני (`bool`)

א. כתבו במחלקה `Truck` את הפונקציה הבונה (`__init__`) המקבלת את מספר הרישוי ואת שם הנהג. הפונקציה מאתחלת את המשאית הפנויה להובלה, עם משקל מרבי של 4,000 קילוגרם ושאינה מיועדת לקירור.

(3 נקודות)

ב. כתבו פונקציה חיצונית המקבלת רשימה של משאיות (רשימת הפניות לאובייקטים מטיפוס `Truck`) ומדפיסה את לוחיות הרישוי של המשאיות המיועדות לקירור שמשקלן לפחות 10,000 קילוגרם.

(4 נקודות)

ג. כתבו פונקציה המקבלת רשימה של משאיות (רשימת הפניות לאובייקטים מטיפוס `Truck`) ומחזירה את **שם הנהג** של משאית שהיא **פנויה** להובלה, **מיועדת לקירור** ומשקל האחסון שלה הוא **הגדול ביותר**.

אם אין אף משאית פנויה שמיועדת לקירור, הפונקציה תחזיר מחרוזת "NONAME".

(5 נקודות)

5. **א.** כתבו פונקציה המקבלת מספר שלם וחיובי `num` ומחזירה את סכום כל מחלקיו החיוביים פרט לו עצמו.

כותרת הפונקציה:

```
def sum_of_proper_divisors(num)
```

לדוגמה:

- עבור המספר 10 הפונקציה תחזיר $8 = 1 + 2 + 5$
- עבור המספר 6 הפונקציה תחזיר $6 = 1 + 2 + 3$
- עבור המספר 12 הפונקציה תחזיר $16 = 1 + 2 + 3 + 4 + 6$

(6 נקודות)

ב. כתבו פונקציה המקבלת מספר שלם חיובי `num`, ומחזירה את אחד מהערכים הבאים המציינים "סוג המספר":

- `'P'` - אם `num` שווה לסכום מחלקיו (מספר משוכלל - Perfect).
- `'D'` - אם `num` גדול מסכום מחלקיו (מספר חסר - Deficient).
- `'A'` - אם `num` קטן מסכום מחלקיו (מספר עתיר - Abundant).

כותרת הפונקציה:

```
def check_number_type(num)
```

הפונקציה חייבת להשתמש בפונקציה שכתבתם בסעיף א'.

(3 נקודות)

ג. כתבו פונקציה המקבלת מספר שלם חיובי `limit` ותו `num_type` ומחזירה רשימה הכוללת את כל המספרים השלמים החיוביים הקטנים מ-`limit` בהתאם לסוג המבוקש `num_type`.

לדוגמה:

עבור המספר `num = 60` וסוג המספר `num_type='A'` הפונקציה תחזיר את הרשימה:

`[12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56]`

כותרת הפונקציה:

```
def all_numbers_by_type(limit, num_type)
```

(6 נקודות)

6. "סיבוב שמאלה של מחרוזת" היא פעולה שבה התו הראשון של המחרוזת נמחק ומועבר לסופה.

לדוגמה: אם `s = "ABCD"` אזי תוצאת הפונקציה היא: "BCDA".

מחרוזת `s2` היא "**גלגול K**" של מחרוזת `s1` אם אפשר לקבל את מחרוזת `s2` באמצעות K סיבובים שמאלה של מחרוזת `s1`.

לדוגמה: אם `s1 = "ABCDE"` אזי `s2="CDEAB"` היא גלגול 2 של מחרוזת `s1`.

א. כתבו פונקציה המקבלת שתי מחרוזות `s1` ו-`s2` ומספר שלם חיובי `k`. הפונקציה תבדוק אם מחרוזת `s2` היא "גלגול K" של מחרוזת `s1`. אם כן - הפונקציה תחזיר `True`, ולא - הפונקציה תחזיר `False`.

כותרת הפונקציה:

```
def is_k_rolling(s1, s2, k)
```

(5 נקודות)

ב. כתבו פונקציה המקבלת שתי מחרוזות `s1` ו-`s2`. הפונקציה תחזיר מספר שלם `k` שעבורו מחרוזת `s2` היא "גלגול K" של מחרוזת `s1`. אם אין שום מספר `k`, הפונקציה תחזיר `-1`.
כותרת הפונקציה:

```
def find_rolling_k(s1, s2)
```

(5 נקודות)

ג. כתבו פונקציה המקבלת רשימת מחרוזות `arr` ומחרוזת נוספת `s`. על הפונקציה לבדוק אם כל המחרוזות שנמצאות ב-`arr` הן "גלגול K" כלשהו של המחרוזת `s`. אם כן - הפונקציה תחזיר `True`, ולא - הפונקציה תחזיר `False`.
כותרת הפונקציה:

```
def are_all_rotations(arr, s)
```

(5 נקודות)

7. נתונה הפונקציה `what` המקבלת רשימה של מספרים שלמים חיוביים `arr` ופרמטר נוסף `exp`. הפרמטר `exp` יכול לקבל ערכים שהם חזקות של 10 (כגון 1, 10, 100, 1000 ...).

```
def what(arr, exp):
    n = len(arr)
    output = [0] * n
    count = [0] * 10

    for i in range(n):
        digit = (arr[i] // exp) % 10
        count[digit] += 1
# **** סיום שלב 1 ****

    for i in range(1, 10):
        count[i] += count[i - 1]
 # **** סיום שלב 2 ****

    for i in range(n - 1, -1, -1):
        digit = (arr[i] // exp) % 10
        output[count[digit] - 1] = arr[i]
        count[digit] -= 1
 # **** סיום שלב 3 ****

    return output
```

א. נתונה הרשימה `arr = [23, 15, 9, 71, 25]`

עקבו אחרי זימון הפונקציה `out1 = what(arr, 1)` ורשמו מה יהיה תוכן הרשימה `out1` אחרי זימון הפונקציה.

יש להראות את תוכן הרשימה `count` אחרי סיום של כל אחד מהשלבים 1-3.

(6 נקודות)

ב. נתונה הרשימה `arr = [23, 15, 9, 71, 25]`

עקבו אחרי זימון הפונקציה `out2 = what(arr, 10)` ורשמו מה יהיה תוכן הרשימה `out2` אחרי זימון הפונקציה.

(6 נקודות)

ג. מה מבצעת הפונקציה `what` באופן כללי?

(3 נקודות)

8. נגדיר "פרח" בתוך מטריצה (רשימה של רשימות) כך:

- ה"פרח" מורכב מחמישה איברים בתת-מטריצה בגודל 3X3. האיבר המרכזי בתת-המטריצה הוא "לב הפרח". ארבעת האיברים הצמודים לו **בפינותיו** (אלכסוניים) הם "עלי הכותרת" של ה"פרח".
- הערך של "לב הפרח" שווה לסכום "עלי הכותרת" של הפרח.
- בכל פרח צריכים להיות בדיוק ארבעה "עלי כותרת".

לדוגמה: במערך הזה יש שני פרחים: אחד עם לב הפרח 6, ושני עם לב הפרח 15.

3		1			
	6				
0		2		8	
			15		
		7		-2	

א. כתבו פונקציה המקבלת מטריצה של מספרים שלמים `arr` ושני מספרים שלמים נוספים המציינים מיקום איבר במטריצה: `row` (שורה) ו-`col` (עמודה). הפונקציה תבדוק אם איבר זה הוא "לב הפרח".

אם כן - הפונקציה תחזיר `True`, ולא - הפונקציה תחזיר `False`.

כותרת הפונקציה:

```
def is_flower(arr, row, col)
```

(4 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. כתבו פונקציה המקבלת מטריצה של מספרים שלמים `arr` ומחזירה רשימה חדשה הכוללת ערכים של כל "לב הפרח" הקיים במטריצה `arr`. אם אין אף "פרח", הפונקציה תחזיר רשימה ריקה.

כותרת הפונקציה:

```
def flower_hearts(arr)
```

(10 נקודות)

ג. מהי סיבוכיות זמן הריצה של כל אחת הפונקציות שכתבתם? הסבירו את תשובתכם.

(3 נקודות)

9. חניון תת-קרקעי משרת כמה חברות היי-טק. לכל חברה מוקצית מכסה מקסימלית של מקומות חניה (מספר הרכבים של החברה אשר יכולים לחנות בו-זמנית), ויש לנהל את הרכבים בהתאם לחברה שאליה הם משויכים ולמכסות המוקצות לה. לשם כך הוגדרו המחלקות הבאות:

המחלקה "רכב חברה" (**`CompanyCar`**) ולה התכונות:

- מספר הרישוי (`license_plate`) - מחרוזת, `str`
- היצרן (`manufacturer`) - מחרוזת, `str`
- שם החברה שהרכב משויך אליה (`company`) - מחרוזת, `str`

המחלקה "מגבלת חברה" (**`CompanyLimit`**) ולה התכונות:

- שם החברה (`company_name`) - מחרוזת, `str`
- המגבלה המקסימלית (`company_limit`) - מספר שלם, `int`

בכל מחלקה הוגדרה פונקציה בונה (`__init__`) המקבלת ערכים לכל התכונות.

א. (1) הגדירו מחלקה בשם "חניון חברות" (**`Parking`**) הכוללת את התכונות הבאות:

- **רשימה של רכבים** (`cars`) - רשימה של עצמים מסוג "**רכב חברה**". הרשימה תכיל את הרכבים הנמצאים כרגע בחניון.
- **רשימה של מקומות חניה** (`parking_spots`) - רשימה של מחרוזות באותו הגודל, המציין את מקום החניה של הרכב המתאים (לדוגמה: "P10", "A4", "Q21").
- **מגבלות חניה** (`limits`) - רשימה של עצמים מסוג `CompanyLimit`, המכילה את המכסות לכל חברה.

(2) כתבו במחלקה `Parking` פונקציה בונה (`__init__`), המקבלת רשימת `limits` מוכנה (עם החברות והמגבלות שלהן), ומייצרת חניון ללא רכבים: רשימות ה-`cars` וה-`parking_spot` יהיו ריקות בתחילה. **קיבולת החניון הכוללת היא 150 מקומות.**

(4 נקודות)

ב. כתבו במחלקה "**חניון חברות**" פונקציה המקבלת **מקום בחניה** (`spot`). הפונקציה תבדוק אם מקום זה פנוי. אם כן - הפונקציה תחזיר ערך `True`, ואם לא - הפונקציה תחזיר ערך `False`. כותרת הפונקציה:

```
def check_spot(self, spot)
```

(5 נקודות)

ג. כתבו במחלקה "**חניון חברות**" פונקציה המקבלת את פרטי הרכב ואת מקום החניה (`spot`) ומנסה לחנות אותו:

- אם החניה `spot` תפוסה על ידי רכב אחר, הפונקציה תדפיס הודעה מתאימה ותחזיר `False`.
- אם החברה כבר ניצלה את המכסה שלה, הפונקציה תדפיס הודעה מתאימה ותחזיר `False`.
- אם המכסה **לא** נוצלה במלואה, הפונקציה תייצר רכב חדש (עצם מסוג `CompanyCar`) עם הפרטים שקיבלה, תוסיף את הרכב ותחזיר `True`. כותרת הפונקציה:

```
def park_car(self, license, manufacturer, company, spot)
```

אפשר להניח שיש מקומות פנויים בחניון, והרכב (עם מספר הרישוי הנתון) אינו נמצא בחניון.

(8 נקודות)

10. בשאלה זו אין קשר בין הסעיפים

א. נתונה הפונקציה `what` המקבלת שני מספרים שלמים חיוביים `a` ו-`b` (גדולים מאפס). (`n=1` הינו ערך ברירת מחדל של `n`, במידה ולא מסופק ערך ל-`n` בעת קריאה לפונקציה).

```
def what(a, b, n=1):
  if a == 1:
    return b == 1
  if n == b:
    return True
  if n > b:
    return False
  return what(a, b, n * a)
```

(1) מה תהיה תוצאת זימון הפונקציה `what(2, 15)`? **יש להראות מעקב.**

(2) תנו דוגמה למספר דו-ספרתי `b` שעבורו תוצאת זימון הפונקציה `what(3, b)` תהיה `True`.

(3) מה מבצעת הפונקציה `what(a, b)` באופן כללי עבור שני מספרים שלמים חיוביים?

(6 נקודות)

ב. נתונה הפונקציה `why` המקבלת מספר שלם `x=None` . `n`, ערך ברירת המחדל של `x`.

```
def why(n, x = None):
  if x is None:
    x = n
  if n > x * x:
    return False
  if n < x * x:
    return why(n, x - 1)
  return True
```

(1) מה תהיה תוצאת זימון הפונקציה `why(10)`? **יש להראות מעקב.**

(2) תנו דוגמה למספר דו-ספרתי `n` שעבורו תוצאת זימון הפונקציה `why(n)` תהיה `True`.

(3) מה מבצעת הפונקציה `why` באופן כללי עבור מספר שלם `n`? **יש להתייחס למקרים שונים של n.**

(6 נקודות)

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ג. נתונה הפונקציה `display` המקבלת מספר שלם:

```
def display(n):
  if n > 0:
    display(n // 10)
    print(n % 10, end="")
    display(n // 10)
```

(1) מה יהיה פלט עבור זימון הפונקציה `display(12)`? **יש להראות מעקב.**

(2) מה יהיה פלט עבור זימון הפונקציה `display(123)`?

(3) מה יהיה פלט עבור זימון הפונקציה `display (1234)`?

(5 נקודות)

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

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il