

מחולל ריבועי קסם

מבוא

צור תוכנית שיוצרת ומאמתת ריבועי קסם בשיטה הסיאמית. ריבוע קסם הוא מערך $n \times n$ מלאה במספרים שלמים חיוביים $(1-n^2)$ כך שסכום המספרים בכל שורה, עמודה ואלכסון זהה.

יש להשתמש בשכלונה של מעבדה 8.

תוכנית ראשית

```
FUNCTION myexercise():
    WHILE true:
        // Get user input
        PRINT welcome message
        PRINT prompt for square size
        size = READ integer from user

        IF size == 0:
            EXIT program

        // Validate input
        IF size is not between 3 and 10 OR size is even:
            CALL handle_invalid_input()
            CONTINUE to next iteration

        // Process valid input
        PRINT confirmation of size

        // Allocate memory for square (size × size integers)
        ALLOCATE square array on stack

        // Generate, display, and verify the magic square
        CALL generate_magic_square(size, square_array)
        CALL print_square(size, square_array)
        CALL verify_square(size, square_array)

        // Clean up and show goodbye message
        PRINT goodbye message
        DEALLOCATE square array from stack
    END WHILE
END FUNCTION
```

פונקציות למימוש

1. Input Validation Function

```
FUNCTION handle_invalid_input():
    // Display error message for invalid size
    // Size must be odd and between 3-10 inclusive
END FUNCTION
```

2. Magic Square Generation (Siamese Method)

יש להקצות את המערך בתוכנית הראשית על ידי הזזה של `sp` בגודל הנדרש. יש להשתמש ב-`fp` בתוך הפונקציה הנקראת כדי לגשת למערך. זאת אומרת, הלולאה הראשית מקצה את המערך, והוא מועבר במחסנית לפונקציה הנקראת. עיין בשקף האחרון של המצגת על פרוצדורות.

```
FUNCTION generate_magic_square(size, square_array):
    // Initialize all cells to 0
    // Start position: row = 0, column = size/2
    // Fill numbers 1 through size^2

    FOR each number from 1 to size^2:
        // Place current number in current position

        // Calculate next position using Siamese method:
        // - Move up one row, right one column
        // - Handle wraparound at edges
        // - If position is already filled, move down from previous position
    END FOR
END FUNCTION
```

השיטה הסיאמית

התחל באמצע השורה העליונה

עבור למעלה וימינה עבור כל מספר חדש

אם אתה עובר מעל השורה העליונה, תעבור לשורה התחתית.

אם אתה עובר את העמודה הימנית ביותר, עבר לעמודה השמאלית ביותר.

אם התא כבר תפוס, מקם את המספר מתחת למספר התפוס.

מומלץ "להתאמן" על נייר לפני התכנון.

Siamese Method Rules:

- Start at the middle of the top row
- Move up and to the right for each new number
- If you go above the top row, wrap to the bottom
- If you go past the rightmost column, wrap to the leftmost
- If the cell is already occupied, place the number below the previous number instead

3. Display Function

יש לוודא יישור עמודות.

```
FUNCTION print_square(size, square_array):
    // Print the magic square in a formatted grid
    // Each number should be padded for proper alignment

    FOR each row:
        FOR each column:
            CALL format_cell(square_array[row][column])
        PRINT newline
    END FOR
END FUNCTION
```

4. Cell Formatting Function

פונקציה שמרפדת את המספר לפי הפורמט רווח ושלוש ספרות

```
FUNCTION format_cell(number):  
    // Convert integer to padded string representation  
    // Format: " xxx " (4 characters total with spaces for alignment)  
    // Handle 1-digit, 2-digit, and 3-digit numbers  
    // Add appropriate leading spaces for alignment  
END FUNCTION
```

5. Verification Function

פונקציית אימות. אם מימשת את האלגוריתם בצורה תקינה, כל המספרים יהיו זהים.

```
FUNCTION verify_square(size, square_array):  
    // Calculate and display sums for verification  
  
    // Calculate row sums  
    PRINT "Row sums: "  
    FOR each row:  
        sum = SUM of all elements in row  
        CALL format_cell(sum)  
    END FOR  
  
    // Calculate column sums  
    PRINT "Column sums: "  
    FOR each column:  
        sum = SUM of all elements in column  
        CALL format_cell(sum)  
    END FOR  
  
    // Calculate diagonal sums  
    PRINT "Diagonal sums: "  
    sum1 = SUM of main diagonal (top-left to bottom-right)  
    sum2 = SUM of anti-diagonal (top-right to bottom-left)  
    CALL format_cell(sum1)  
    CALL format_cell(sum2)  
  
    // Optional: Check if all sums are equal and display verification result  
END FUNCTION
```

אתגרי תכנות

ניהול זיכרון

- הקצאת שטח עבור מערך $n \times n$ באופן דינמי
- ניהול נכון של שטח המחסנית עבור משתנים מקומיים
- טפל בניקוי (החזרת מחסנית, שחזור אוגרים) אחר השלמת פונקציה

יישום אלגוריתם

- יישם את השיטה הסיאמית בצורה נכונה עם לוגיקה נכונה
- טפל במקרי קצה כאשר המיקום הבא כבר תפוס
- וודא שריבוע הקסם מלא במספרים עוקבים עד n^2

עיצוב קלט/פלט

- יצירת הנחיות והודעות ידידותיות למשתמש
- עיצוב מספרים עם מרווח עקבי עבור תצוגת נכונה
- טיפול באימות קלט בצורה נכונה

לוגיקת אימות חישוב

- סיכום ביעילות באמצעות לולאות
- הצג תוצאות בתבנית ברורה ומאורגנת
- אופציונלי: ליישם אימות אוטומטי שבודק שכל הסכומים שווים

דוגמה של הרצה

```
Welcome to the RISC-V Magic Square Generator!

Enter the size of the square: 3

Making square of size: 3

  2  7  6
  9  5  1
  4  3  8

Row sums:   15   15   15
Column sums: 15   15   15
Diagonal sums: 15   15

Thank you for using this program!
```

הערות ליישום

קבוע הקסם (סכום כל שורה/עמודה/אלכסון) עבור ריבוע קסם $n \times n$ הוא: $\frac{n(n^2+1)}{2}$. ניתן ליצור רק ריבועים בגודל אי-זוגי בשיטה הסימטרית הבסיסית בדוק עם גדלים שונים (3, 5, 7, 9) כדי לוודא שהאלגוריתם שלך פועל כהלכה שקול מקרי קצה וטיפול בשגיאות לאורך ההטמעה שלך