

שאלות מעשיות: כתיבת והרצת קוד אסמבלי למIPS:

עבור שתי השאלות הבאות צורף לתרגיל שלד קוד אסמבלי למIPS בו חסר מימוש לחלק מהפונקציות. בשני המקרים יש להוסיף את תוכן הפונקציות לקבצים הקיימים ולצרפם בעת הגשת התרגיל. שימו לב – מותר להשתמש רק בפקודות שראינו בשיעור או בתרגול.

שאלה 4:

לתרגיל הבית צורף קובץ בשם fibo.asm. קובץ זה מהווה שלד להרצת תוכנית המקבלת מספר כקלט מהיורז ומדפיסה את האיבר המתאים לו בסדרת פיבונאצ'י. לדוגמא, עבור קלט של 20, התוכנה מדפיסה את האיבר 20 בסדרת פיבונאצ'י (6765). לצורך חישוב המספרים הקוד קורא לפונקציה fib, אך מימושה חסר. ממשו את הפונקציה החסרה, תוך הקפדה על עקרונות הcalling conventions. תזכורת: את סדרת פיבונאצ'י ניתן לחשב בעזרת קוד c הבא:

```
int fibonacci (int n) {
    if (n==0) return 0;
    if (n==1) return 1;
    return fibonacci(n-1) + fibonacci(n-2);
}
```

שאלה 5:

לתרגיל הבית צורף קובץ בשם palindrome.asm. קובץ זה מהווה שלד להרצת תוכנית המקבלת מהיורז מחרוזת כקלט ובודקת האם המחרוזת היא פלינדרום (כלומר, האם קריאתה מימין לשמאל או משמאל לימין זהה). לצורך השאלה, גם מחרוזת באורך 0 הינה פלינדרום. לצורך חישוב המספרים, הקוד קורא למספר פונקציות אותן יש לממש תוך הקפדה על עקרונות הcalling conventions.

- א. ממשו את הפונקציה get_length המקבלת ברגיסטר \$a0 את המיקום בזכרון של תחילת המחרוזת ומחזירה את אורכה בביטים. לצורך כך הניחו כי לכל קלט שמתקבל מהיורז נוסף באופן אוטומטי תו סיום השורה: '\n', אך האורך המוחזר אינו צריך לכלול אותו (כלומר, אורכה של המחרוזת 'abc' ינתן על פי 3 תווים ולא 4). השורה הראשונה במימוש הנתון של הפונקציה מציבה את תו זה ברגיסטר \$t0. שימו לב כי המחרוזת בשאלה נתונה כמערך (בו כל תו מהווה מילה, כלומר 4 בתיים).
- ב. ממשו את הפונקציה is_pali_loop המקבלת ברגיסטר \$a0 את המיקום בזכרון של תחילת המחרוזת וב1 \$a1 את אורכה בביטים. הפונקציה תחזיר 1 במידה והמחרוזת היא פלינדרום ו0 אחרת. על הפונקציה להיות ממומשת על ידי לולאות, וללא קריאות רקורסיביות.
- ג. ממשו את הפונקציה is_pali_recursive המקבלת קלט זהה ומחזירה פלט זהה לסעיף ב'. על הפונקציה להיות ממומשת על ידי רקורסיה, ללא שימוש בלולאות.

😊 בהצלחה! 😊