

לד (LED) היא רכיב אלקטרוני שפולט אור בעת מעבר זרם חשמלי. בתרגיל זה נתכנת דימוי (simulation) של שליטה על לדים ע"י המשתמש. נתון שכל ביט באוגר (s0) מייצג מצב של LED אחד (1 = ON, 0 = OFF). במצב ההתחלתי, s0 שווה לאפס, כלומר כל 32 הלדים כבויים. ביט 0 מייצג את LED0 וכו'. המשתמש מקליד פקודה של תו בודד. יש פקודות שעבורם המשתמש גם יקליד מספר שמייצג מיקום (0-31) של ביט מתוך האוגר. אחרי כל קליטה של פקודה, התוכנית תדפיס את מצב של כל הביטים - נקודה (.) אם הוא במצב של OFF, סולמית (#) אם הוא במצב של ON.

הפקודות

1. 's' (Set bit) → Turn ON the LED at numeric position following the s.
2. 'c' (Clear bit) → Turn OFF the LED at numeric position following the c.
3. 't' (Toggle bit) → Flip the LED state at numeric position following the t.
4. 'l' (Shift Left) → Shift all LEDs **left** by one.
5. 'r' (Shift Right) → Shift all LEDs **right** by one.
6. 'b' (Rotate Right) → Shift all LEDs **right** by one, and move the LSB to the MSB
7. 'd' (Rotate Left) → Shift all LEDs **left** by one, and move the MSB to the LSB

1. תכתוב פרוצדורה **led_control** שמקבלת שלושה פרמטרים:

a0 → Current LED register (bits representing LED states).

a1 → Command (s, c, t, l, r, b, or d).

a2 → Bit position (0-31), only for s, c, t.

הפרוצדורה תחזיר ב-a0 את הביטים בהקדם לפקודה.

2. תכתוב פרוצדורה **print_led** שמדפיסה את המצב של הביטים שמייצגים את הלדים לפי דוגמת הרצה למטה.

3. תכתוב את ה-main של התוכנית. ה-main קולט תו מהמשתמש, ואם יש צורך הוא קולט מספר, וקוראת ל-**led_control** ול-**print_led** וחוזר לראש כדי לקלוט עוד פקודה.

דוגמת הרצה:

```
.....
s3
.....#...
s5
.....#.#...
t2
.....###..
t2
.....#.#...
c3
```

```
.....#.....
s1
.....#...#.
r
.....#...#
r
.....#...
s0
.....#..#
b
#.....#..
b
.#.....#.
b
..#.....#
b
#..#.....
d
..#.....#
d
.#.....#.
d
#.....#..
d
.....#..#
```

בהצלחה!