

## מחשבון פשוט

כתוב תוכנית ליישום מחשבון פשוט.

### דרישות (Requirements)

- קלט: התוכנית תציג >> כסימן לקלוט מספר שלם או אופרטור.
- מספר יכול להיות חיובי או שלילי.
- אופרטורים הם (+, -, \*, =).
- עיבוד:
  - אחרי קלט של =, יש להדפיס את התוצאה.
  - עדיפות אופרטורים לפי סדר קליטה.
  - יש לטפל במספרים חיוביים ושליליים.
  - יש לזהות גלישת מספרים שלמים (overflow) ולהדפיס הודעת שגיאה.
  - אין צורך לטפל במקרים שחסר אופרטור או מספר.
- פלט: הדפס את התוצאה או הודעת שגיאה.
- לולאה: התוכנית צריכה להמשיך לבקש קלט.

### מימוש (Implementation)

#### 1. קלט (Read Input)

- השתמש בקריאת מספר (ecall 5) או קריאת תו (ecall 12)
- אחסן את הקלט ברגיסטרים.

#### 2. בצע חישוב (Calculate)

- בצע חיבור, חיסור וכפל.
- טפל בגלישה

#### 3. תוצאת הדפס (Print)

- הדפס Result: והתוצאה, או הודעת שגיאה.
- לחזור לשלב 1.

### דוגמת הרצה

```
>> 5
>> +
>> 6
>> *
>> 5
>> =
Result: 55
>> 1000000000
>> *
>> 1000000000
>> =
Overflow error!
```

## בדיקת גלישה

חיבור (יכול לקרות רק עם שני המספרים בעלי אותו סימן, והסימן של התוצאה שונה מסימן המספרים)

```
# Overflow detection for addition, t1 = a, t2 = b
addi t0, t1, t2      # t0 = t1 + t2 (result)
xor t3, t1, t2       # t3 = t1 ^ t2 (check if signs are different)
blt t3, zero, no_overflow_add # If signs are different, no overflow
xor t4, t1, t0        # t4 = t1 ^ t0 (check if result sign differs from a)
bge t4, zero, no_overflow_add # If result sign same as a, no overflow
# Handle overflow
j end_add
no_overflow_add:
# Continue normal execution
end_add
```

עליך לממש בדיקה גם עבור חיסור!

## כפל

יש להשתמש ב-MULH לקבל את HI, וב-MUL כדי לקבל את LO.

במידה ותוצאת הכפל הינה חיובית (סימנים זהים) אזי אם רגיסטר HI ערכו שונה מ-0 פירושו שהתוצאה גדולה יותר מ-32 ביט. אם רגיסטר HI כולו '0' פירושו שגודל התוצאה הינו 32 ביט.

במידה ותוצאת הכפל הינה שלילית (סימנים שונים) אזי אם רגיסטר HI ערכו שונה מ-1 פירושו שהתוצאה גדולה יותר מ-32 ביט. אם רגיסטר HI כולו '1' פירושו שגודל התוצאה הינו 32 ביט.

```
mulh t3, t1, t2 # Upper 32 bits
mul t0, t1, t2 # Lower 32 bit
srai t4, t0, 31 # Sign-extend lower 32 bits
bne t3, t4, overflow_detected
```

הקוד הנ"ל בודק האם HI שונה מביט 31 של ה-LO.

בשיעור הוזכרו מקרים נוספים של גלישות ומקרי קצה. לדוגמא,

$$(-1) * (0x40000000) = -1073741824 \text{ (0xFFFFFFF0000000)}$$

אין צורך לתמוך במקרים הנ"ל, רק במקרים למעלה. אם יש זיהוי של גלישה, יש להדפיס שגיאה ולהמשיך את הלולאה.

בהצלחה!