

מטלה 5 – קיץ תשפא

למען הסר ספק, זוהי מטלת MIPS. מי שיפתור ב assembly של HACK לא יקבל ניקוד כלל.

כל סעיף הינו קובץ asm נפרד. שאלות התרגום הינן קובצי txt בלבד. לא asm.

בשאלות התרגום יש להגיש אך ורק את התרגום. ללא השאלה הרשומה בדף המטלה. ללא הערות. ללא דרך.

כל הקבצים ביחד יכווצו לזיפ ששמו כמספר הזהות שלכם, עם 5_ בסוף.

יש לכתוב הערות בקוד, לפחות כל שורה שלישית. קוד ללא הערות **שאינו תקין** לא יקבל ניקוד כלל.

תזכורת: ציון מטלה זו מהווה 50% מציון המטלות.

ציון המטלה:

מתוך המטלה ייבדקו 3-4 שאלות שיהוו 50% מציון המטלה. שאר הסעיפים יקבלו ניקוד מלא כל עוד הם קיימים ורלוונטיים (אני כן פותח להסתכל, אבל לא בודק לעומק).

1. הכניסו את מספר הזהות שלכם כך ששמונה הספרות הראשונות יופיעו (כפי שהן! למרות שהסימולטור מציג ערכים בהקסה-דצימאלי!) באוגר t4 וספרת הביקורת (הספרה ה LSB של מספר הזהות) תירשם בספרה השלישית מימין (כלומר, ה Nibble שאינו הכי קטן, אלא אחד מעל) של אוגר s1.

2. יהיו המערכים

$$\begin{aligned}A &= [1, 3, 5, 7, 9, 11, 13] \\B &= [1, 4, 9, 16, 25, 36, 49] \\C &= [1, -1, 1, -1, 1, -1, 1]\end{aligned}$$

כתבו קוד שמחשב את

$$\sum_{i=1}^7 A_i \cdot B_i \cdot C_i$$

אם התוצאה יש לרשום באוגר t2. אם התוצאה חורגת מ 32 ביטים, את הביטים העליונים יש לרשום באוגר t1.

אין להשתמש בתאי הזיכרון כדי לשמור את המערכים! יש להשתמש באוגרים, ולשנות את ערכם בהתאם בתוך לולאה.

3. תרגמו את הקוד הבא ל Assembly. השתמשו באוגר \$s6 עבור המשתנה Num.

Num:=(The Maximal digit in your ID) - 1

(למען הסר ספק- המשתנה Num מקבל את הספרה המקסימלית במספר הזהות שלך, פחות 1)

```
Switch(Num){
  Case 1:
  {
```

```

        Num+=0x20;
        Break;
    }
    Case 2:
    {
        Num*=8;
        Break;
    }
    Case 3:
    {
        Num=(5*Num+20)
        Break;
    }
    Default:
    {
        Num = Num div 3
    }
}

```

4. כתבו קוד המכניס את הערכים 1,8,64,512,4096 לתאי הזיכרון
 $0x10010000, 0x1001000C, 0x10010018, 0x10010024, 0x10010030$.

יש לבצע את ההכנסה בלולאה.

5. תרגמו את הקוד הבא לאסמבלי. הקפידו על שימוש נכון במשתנים וקריאה נכונה לפונקציות. את התוצאה יש לשמור באוגר \$s2.

```

Main:
{
    Func1(10,0x10,0x12)
}
Func1(a,b,c)
{
    sum=a+b-c
    sum+=Func2(sum,a,b,c)
    Return sum
}
Func2(sum,a,b,c)
{
    sum=sum*(a+b+c)
    Return (sum+1)
}

```

6. נתון הקוד הבא. תרגמו אותו להקסה-דצימאלית (הגשת הפתרון בפורמט בינארי תידחה).

נתוני עזר:

$funct(add) = 32, \quad funct(sub) = 34, \quad opcode(beq) = 4,$

$opcode(bne) = 5,$ $opcode(j) = 2,$ $opcode(addi) = 8,$
 $opcode(sw) = 43,$ $opcode(lw) = 35,$ $funct(slt) = 42$
 $funct(sll) = 0$

0x00400000: add \$t2, \$s7, \$a3
 0x00400004: label1: sub \$s2, \$v1, \$1
 0x00400008: bne \$s1, \$t3, label2
 0x0040000C: beq \$t0, \$s0, label1
 0x00400010: addi \$t8, \$t7, 1234
 0x00400014: label2: j label3
 0x00400018: sw \$s5, 60(\$t0)
 0x0040001C: j label2
 0x00400020: label3: slt \$k0, \$v0, \$a1
 0x00400024: sll \$s1, \$s7, 3

7. נתון הקוד הבא בפורמט הקסה-דצימאלי. תרגמו אותו לקוד *assembly*.

0x20101234
 0x00108400
 0x02058020
 0x0153482A
 0x11200003
 0x01378804
 0x0327A825
 0x08100003
 0x01A69022

נתונים נוספים:

$funct(sll) = 0$ ובפקודה זו האוגר rs מאופס (הראשון הוא rd והשני הוא rt , ואילו rs לא קיים).

$funct(sllv) = 4$ ובפקודה זו הסדר הינו rd, rt, rs .

$funct(and) = 36,$ $funct(or) = 37,$ $funct(xor) = 38,$ $funct(nor) = 39$

בנוסף, כדאי להיעזר בנתונים שנרשמו בשאלה הקודמת.

שמות התוויות חסרי חשיבות. כדי לקבל פתרון "אחיד" קראו לתווית הראשונה *label1* ולתווית השנייה *label2*.

כזכור, אין לכתוב כלל הערות בקוד שפענחתם. רק את הקוד עצמו.

8. המירו את הפונקציה הרקורסיבית הבאה לאסמבלי:

```
Hanoi(n){  
    If n==1  
        Return 1  
    Else    return(2*Hanoi(n-1)+1)
```

יש להשתמש במחסנית כפי שראינו בתרגול.

יש לבנות אך ורק את הפונקציה הרקורסיבית, ללא *main* שקורא לה!

טסטים למטלה:

1. תראו את מספר הזהות שלכם ב *Mars* במקום הנכון.
2. חישוב – השוו תוצאה עם סטודנטים אחרים.
3. הזינו כל *Num* אפשרי בשורה הראשונה של הקוד כדי לוודא שהוא פועל כראוי. שימו לב שצריך להחזיר את האתחול של *Num* לפי מה שמוגדר בשאלה.
4. הסתכלו בזיכרון של ה *MARS* (מופיע למטה כשמריצים תוכנה) וודאו שהערכים הללו אכן נכנסו למקומות הנכונים.
5. חישוב – השוו תוצאה עם סטודנטים אחרים (ודאו שהקוד שלכם מסיים את הריצה ולא נתקע!).
- 6+7. חשבו איך אפשר להיעזר בתוכנת *Mars* כדי לבדוק את עצמכם.
8. כתבו *Main* הקורא לפונקציה (אל תשכחו לדלג עליה כשחוזרים!) והזינו ערכים עבור *n* כדי לבצע בדיקה. אל תשכחו למחוק את ה-*Main* לפני ההגשה.