

את"ם – תרגיל בית מס' 3 סמסטר אביב תשע"ו

תאריך פרסום: 15.12.2015 תאריך הגשה: 31.12.2015 (בשעה 23:55)
מתרגל אחראי על התרגיל: אביב סגל (segaviv@cs.technion.ac.il)

- הגשה בזוגות בלבד לתא ההגשה של הקורס ובאמצעות הגשה אלקטרונית.
- שאלות על התרגיל יש להפנות לאביב segaviv@cs.technion.ac.il.
- הגשות באיחור יש לתאם עם אביב לפני מועד ההגשה הכללי.
- ערעורים והגשות באיחור יש להגיש לתא הדואר של אביב.
- אין להגיש לתא הקורס לאחר מועד ההגשה.

נושא התרגיל: שגרות + מבנים.

בתרגיל זה שני חלקים:

- חלק א' מכיל שאלה אחת, עליכם לענות על השאלה בכתב ולהגיש לתא הקורס (יש להדפיס את טופס התרגיל ולענות על גביו).
- חלק ב' דורש כתיבת קוד בשפת האסמבלי של PDP-11, כפי שנלמד בהרצאות ובתרגולים. את הקוד יש לכתוב בקובץ ex3.s11, ולהגיש הדפסה של התייעוד החיצוני בלבד, כמפורט בהמשך, לתא הקורס. כדאי לקרוא באתר הקורס ב-FAQ על רמת התייעוד הנדרשת. כמו כן יש להגיש את הקובץ ex3.s11 zip כפי שמצוין באתר).
- קריאת ה-FAQ של התרגיל היא חובה.

חלק יבש

ענו על השאלות שבעמוד הבא על גבי טופס התרגיל, והגישו לתא הקורס.

עיינו בתוכנית הבאה וענו של השאלות שאחריה.

1.	.	=	torg+1000		
2.	main:	mov	pc,	sp	
3.		sub	#6,	sp	
4.		mov	#root,	-(sp)	
5.		jsr	pc,	g	
6.		tst	(sp)+		
7.		mov	(sp)+,	res1	
8.		mov	(sp)+,	res2	
9.		halt			
10.	g:	clr	4(sp)		
11.		mov	#1,	6(sp)	
12.		tst	2(sp)		
13.		beq	gexit		
14.		mov	r3,	-(sp)	
15.		mov	r4,	-(sp)	
16.		mov	6(sp),	r3	
17.		mov	(r3),	in_f	
18.		jsr	r5,	f	
19.		mov	r4,	10(sp)	
20.		cmp	-(sp),	-(sp)	
21.		mov	2(r3),	-(sp)	
22.		jsr	pc, g		
23.		jsr	pc, h		
24.		mov	4(r3),	(sp)	
25.		jsr	pc,	g	
26.		jsr	pc,	h	
27.		add	#6,	sp	
28.		mov	(sp)+,	r4	
29.		mov	(sp)+,	r3	
30.	gexit:	rts	pc		
31.	h:	tst	6(sp)		
32.		bne	h1		
33.		clr	22(sp)		
34.	h1:	cmp	4(sp),	20(sp)	
35.		ble	hexit		
36.		mov	4(sp),	20(sp)	
37.		clr	22(sp)		
38.	hexit:	rts	pc		
39.	f:	mov	in_f,	-(sp)	
40.		clr	r4		
41.	floop:	asl	(sp)		
42.		bcc	f1		
43.		inc	r4		
44.	f1:	tst	(sp)		
45.		bne	flop		
46.		tst	(sp)+		
47.		rts	r5		
48.	.	=	torg+3000		
49.	root:	.word	27, le1, 0		
50.	le1:	.word	11, 0, 0		
51.	in_f:	.blkw	1		
52.	res1:	.blkw	1		
53.	res2:	.blkw	1		

1. עבור כל אחד מהפרמטרים אותם מקבלת השגרה **f**, רישמו בטבלה הבאה מהו שטח ההעברה שלו, וציינו האם הוא משמש לקלט, לפלט או גם לקלט וגם לפלט, וכן האם הוא מועבר לפי ערך או לפי כתובת. אם הפרמטר מועבר דרך אוגר, ציינו מיהו האוגר. שימו לב, כי ייתכן שתשארנה בטבלה שורות ריקות.

פרמטר מספר	שטח העברה	קלט / פלט / שניהם	ערך / כתובת
1			
2			
3			
4			

2. תארו בקצרה מה מבצעת השגרה **f**. (תשובה מעל 10 מילים לא תתקבל)

3. מה יקרה אם נשנה את ההוראה שבשורה 41 להוראה **asr (sp)**? סמנו תשובה אחת נכונה.

- השגרה **f** תעבוד באותו אופן לכל קלט.
- הלולאה בשגרה **f** תסתיים תמיד אחרי איטרציה אחת.
- עבור חלק מהקלטים השגרה **f** תעבוד באותו אופן ועבור חלק מהקלטים היא תחזיר תוצאה שונה.
- עבור חלק מהקלטים השגרה **f** תעבוד באותו אופן ועבור חלק מהקלטים היא תיכנס ללולאה אינסופית.
- הלולאה בשגרה **f** תהיה אינסופית לכל קלט.
- אף אחת מתשובות א' – ה' אינה נכונה.

4. חיזרו על סעיף 1 עבור השגרה **g**:

פרמטר מספר	שטח העברה	קלט / פלט / שניהם	ערך / כתובת
1			
2			
3			
4			

5. תארו את תוכן המחסנית מיד לפני הביצוע הראשון של ההוראה בשורה 22. ניתן לכתוב ביטוי מהצורה "הכתובת של שורה 15" וניתן להשתמש בתוויות. הניחו שבתחילת התוכנית תוכן של אוגר i הוא $i < 6$. במידה שקיים במחסנית ערך לא ידוע, כיתבו "לא ידוע" במקום המתאים.

תוכן (מספר אוקטאלי)	כתובת
	754
	756
	760
	762
	764
	766
	770
	772
	774
	776
010706	1000

6. מלאו בטבלה הבאה את הפלט שתחזיר השגרה **h** עבור כל אחד מהקלטים (נתונים בבסיס דצימלי) הבאים לשגרה.

הקלט ב 6(sp)	הקלט ב 4(sp)	הקלט ב 20(sp)	הפלט ב 20(sp)
1	10	4	
1	5	9	
0	8	2	

7. השלימו את הטענה הבאה כך שתהיה נכונה (יש להקיף בעיגול את האפשרות המתאימה מתוך הסוגריים).

השגרה **h** מאפסת את 22(sp) כאשר (בדיוק אחד / לפחות אחד / שני) התנאים הבאים מתקיים/מתקיימים:

(I) הערך ב 6(sp) הוא (0 / שונה מ-0 / גדול מ-0 / קטן מ-0).

(II) הערך ב 4(sp) (גדול / שווה או גדול / קטן / שווה או קטן) מהערך ב 20(sp).

8. כמה פעמים תיקרא השגרה **g** במהלך התוכנית? _____

9. מה יהיה הערך של **res1** בסוף התוכנית? _____

10. מה יהיה הערך של **res2** בסוף התוכנית? _____

11. לפניכם מספר שינויים בקוד (כל שינוי עומד בפני עצמו ואינו תלוי באחרים). לכל שינוי, כיתבו מה יהיו ערכי המילים המצוינות ע"י התוויות **res1**, **res2** בתום ריצת התוכנית (כיתבו בבסיס עשרוני).

res2	res1	שינוי	
		שורות 49-50 הוחלפו ב- root: .word -1, le1, ri1 le1: .word 5, 0, 0 ri1: .word -5, le2, 0 le2: .word 6, 0, 0	א.
		שורות 49-50 הוחלפו ב- root: .word 17, le1, ri1 le1: .word 5, 0, 0 ri1: .word 73, le2, 0 le2: .word 6, 0, 0	ב.
		שורה 40 הוחלפה ב- mov #20, r4 שורה 43 הוחלפה ב- dec r4	ג.

12. מחליפים את שורות 49-50 בשורות הבאות :

```

root: .word 504, le1, ri1
le1:  .word ____, le2, 0
ri1:  .word 60,  0,  0
le2:  .word 202, 0,  0

```

השלימו את הערך החסר כך שבתום ריצת התכנית הערך ב- **res2** יהיה 1.

13. שורה 35 הוחלפה ב: **blt hexit** .

מלאו את הקלט הבא באופן כזה שהערך של **res2** בסוף ריצת התכנית שונה מהערך שהיה מתקבל בריצה המקורית :

```

root: .word 43, le1, ri1
le1:  .word ____, 0,  0
ri1:  .word 22, le2, 0
le2:  .word 1,  0,  0

```

חלק רטוב (Sliding Puzzle)

תזכורת מתרגיל בית 2

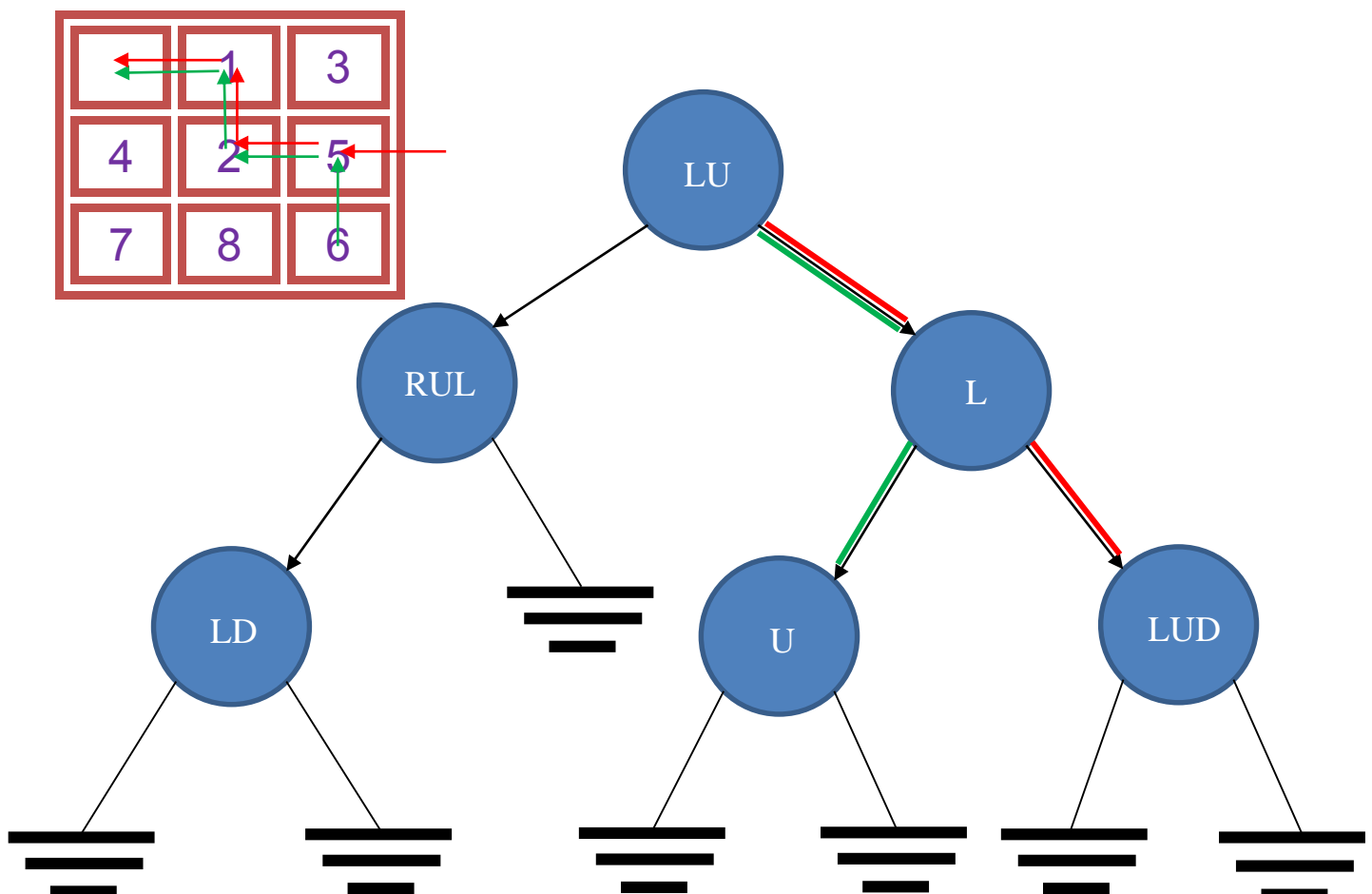
ה Sliding Puzzle היא חידה המורכבת מלוח בגודל $m \times n$ המכיל את המספרים 1 עד $m \cdot n - 1$ ומשבצת אחת ריקה. פתרון החידה מתקבל ע"י סידור המספרים לפי האינדקסים של המערך (כך שהמספר i יהיה באינדקס i) ע"י הזזה בכל שלב של מספר הנמצא בסמוך למשבצת הריקה לתוכה.

בתרגיל בית 2 כתבתם תוכנית המקבלת לוח התחלתי וסדרת מצבים, בודקת האם סדרת המצבים היא חוקית והאם היא פותרת את החידה. בתרגיל זה תתבקשו להרחיב את התרגיל הקודם ע"י מימוש של שגרות ושימוש במבנה נתונים.

תיאור התרגיל

משימה 1

עליכם לכתוב אלגוריתם המחפש סדרת מהלכים בעץ בינארי הפותרת את ה Sliding Puzzle. כלומר, בכל צומת בעץ הבינארי תישמר סדרת מהלכים ועליכם למצוא האם קיים מסלול מהשורש לעלה כלשהו כך שבצומת סדרת המהלכים השמורות בצמתים שבמסלול (לפי סדר הצמתים במסלול) יביא לפתרון של החידה. לדוגמה, בהינתן העץ והמצב ההתחלתי הבאים, המסלול המסומן בירוק מסמן סדרת מהלכים הפותרת את החידה בעוד המסלול המסומן באדום מסמן סדרת מהלכים לא חוקית.



הפלט שייכתב לתווית Output, בדומה לתרגיל הקודם, יהיה :
'L' (Legal) אם כל סדרות המהלכים בעץ חוקיות אך אין סדרה הפותרת את החידה (אין מסלולים ירוקים או אדומים בעץ).

'S' (Solved) אם קיים מסלול בעץ המגדיר סדרת מהלכים הפותרת את החידה.
'I' (Illegal) אם קיימת סדרת מהלכים לא חוקית בעץ ואין סדרת מהלכים הפותרת את החידה.

למשל, עבור הדוגמה מלמעלה הפלט שצריך להיכתב בתווית Output הוא 'S'.

בנוסף, אם קיימת סדרת מהלכים הפותרת את החידה יש לכתוב אותה במערך המוקצה בתווית Moves ולכתוב את אורכה לתוך התווית MovesLen.

משימה II

במשימה זו אתם נדרשים לממש אלגוריתם המוצא עבור צומת כלשהו בעץ את הלוח ההתחלתי שפתרונו מתקבל ע"י ביצוע סדרת המהלכים במסלול מהשורש לצומת.
לדוגמה, עבור הבן השמאלי של שורש העץ בדוגמה ממשימה I, סדרת המהלכים במסלול תהיה LURUL והיא פותרת את הלוח הבא :

1		2
4	6	3
7	5	8

משימה III

כיתבו אלגוריתם המקבל לוח התחלתי וסדרת מהלכים חלקית ומשלים את סדרת המהלכים לסדרה הפותרת את הלוח.

לדוגמה, עבור הלוח ההתחלתי ממשימה II וסדרת המהלכים החלקית L_RU_ האלגוריתם יחזיר את הסדרה LURUL הפותרת את הלוח.

אם לא קיימת השלמה של הסדרה לסדרה הפותרת את הלוח יש להשאיר את המקומות החסרים בסדרה ריקים (כאשר 0 מייצג מקום ריק).

קלט התכנית

- **Width** – תווית המציינת מילה בזיכרון המכילה את מספר העמודות (n).
- **Height** – תווית המציינת מילה בזיכרון המכילה את מספר השורות (m).
- **InitialBoard** – תווית המציינת מערך של $m \times n - 1$ בתים בזיכרון המכיל את הלוח ההתחלתי עבור משימות I ו II.
- **MovesTree** – תווית המציינת מערך של חמש מילים בזיכרון המייצג את שורש העץ הבינארי של המהלכים.
- **Board2** – תווית המציינת מערך של $m \times n - 1$ בתים בזיכרון המכיל את הלוח ההתחלתי עבור משימה III.
- **ParLen** – תווית המציינת מילה המכילה את אורך סדרת המהלכים החלקית.
- **ParMoves** – תווית המציינת מערך באורך **ParLen** בתים המכיל את סדרת המהלכים החלקית.

בכל צומת בעץ שמורות **חמש** מילים המכילות את המידע הבא :

- המילה הראשונה מכילה את אורך סדרת המהלכים השמורה בצומת.
 - המילה השניה מכילה את כתובת סדרת המהלכים של הצומת.
 - המילה השלישית מכילה כתובת שלתוכה ייכתב הלוח שנמצא במשימה II עבור הצומת. אם ערך המילה הוא אפס, אין להפעיל את האלגוריתם ממשימה II עבור הצומת.
 - המילה הרביעית מכילה מצביע לבן השמאלי (או אפס אם אין לצומת בן שמאלי).
 - המילה החמישית מכילה מצביע לבן הימני (או אפס אם אין לצומת בן ימני).
- שימו לב: הנחות לגבי תקינות הקלט מופיעות בסוף מסמך זה, תחת "הערות נוספות". אנא קיראו אותן בעיון.

פלט התוכנית

- **Output** – תווית המציינת בית בזיכרון שלתוכו ייכתב פלט האלגוריתם ממשימה I.
- **Moves** – תווית המציינת מערך בזיכרון שלתוכו ייכתב סדרת המהלכים הפותרת את הלוח במשימה I.
- **MovesLen** – תווית המציינת את אורך סדרת המהלכים הפותרת את הלוח. במידה ואין כזו, התווית תכיל את הערך אפס.
- **ParMoves** – תווית הקלט משמשת גם כתווית פלט. האלגוריתם ממשימה III ישלים את המקומות החסרים במערך (המכילים את הערך 0) כך שתתקבל סדרה הפותרת את הלוח **Board2**.

דוגמה לקלט אפשרי:

```

Width:      .word 3
Height:     .word 3
InitialBoard: .byte 2, 5, 3, 4, 6, 9., 7, 8.
.even
MovesTree:  .word 2, Mv0, 0, node1, node3      ; The root of the tree

node1:      .word 3, Mv1, n1board, node2, 0      ;
node2:      .word 2, Mv2, 0, 0, 0                ; The other nodes
node3:      .word 1, Mv3, 0, node4, node5         ; of the tree
node4:      .word 1, Mv4, 0, 0, 0                ;
node5:      .word 3, Mv5, n2board, 0, 0          ;

Mv0:        .ascii<LU>                          ;
Mv1:        .ascii<RUL>                         ;
Mv2:        .ascii<LD>                          ; The list of moves
Mv3:        .ascii<L>                           ; at each node
Mv4:        .ascii<U>                           ;
Mv5:        .ascii<LUD>                         ;
.even

; Input for task 3
Board2:     .byte 1, 3, 6, 4, 8., 5, 7, 9.
.even
ParLen:     .word 5
ParMoves:   .byte 'L, 0, 'R, 'U, 0

```

בצמתים node1 ו node5 ערך המילה השלישית שונה מ 0, לכן יש להריץ עבורם את האלגוריתם ממשימה II.

הפלט שייכתב לתוויות Output, n2board, n1board, Moves ו MovesLen יהיה:

```

Output:     .byte 'S'
n1board:    .byte 1, 3, 6, 4, 8., 5, 7, 9.
n2board:    .byte 0, 0, 0, 0, 0, 0, 0, 0
Moves:      .byte 'L, 'U, 'L, 'U
MovesLen:   .word 4

```

ערכי המערך שבתווית ParMoves בסוף ריצת התוכנית יהיו:

```

ParMoves:   .byte 'L, 'U, 'R, 'U, 'L

```

שגרות

בתוכנית שלכם **עליכם** לממש מספר שגרות שיפורטו להלן. לכל שיגרה מוסבר תפקידה והממשק שלה (מהו אוגר הקישור, כיצד היא מקבלת ומחזירה פרמטרים וכדומה). הקפידו לממש את הממשק **במדויק**. אסור לשגרות להסתמך על ערכו של אף משתנה גלובאלי אלא אם נאמר אחרת. ניתן להשתמש במערך עזר לצורך מימוש השגרות.

שם השיגרה	תפקיד השיגרה	אוגר קישור	פרמטרים ושטח העברתם
copyArr	מקבלת כתובת מערך מקור, כתובת מערך יעד ואורך ומעתיקה את איברי מערך המקור למערך היעד.	pc	קלט: כתובת מערך המקור, כתובת היעד ואורך המערך (בבתיים) מועברים במחסנית בסדר הנ"ל (כלומר בראש המחסנית יימצא אורך המערך). פלט: העתקת איברי מערך המקור למערך היעד.
makeMovs	מקבלת את כתובת הלוח, כתובת סדרת המהלכים, אורך סדרת המהלכים ואת מימדי הלוח m ו n ומפעילה את סדרת המהלכים על הלוח. בנוסף, השגרה תבדוק אם סדרת המהלכים חוקית.	r5	קלט: כתובת הלוח, כתובת סדרת המהלכים ואורך סדרת המהלכים מועברים inline בסדר הנתון (כתובת הלוח מועברת ראשונה). ממדי הלוח מועברים בשטח המשותף Width ו Height. פלט: שינוי הלוח בהתאם לסדרת המהלכים. בנוסף, תחזיר במחסנית את הערך 1 במידה וסדרת המהלכים חוקית ו 0 אחרת.
checkSol	מקבלת את כתובת הלוח ואת ממדי הלוח ומחזירה האם הלוח נמצא במצב הסופי (הפתור).	pc	קלט: כתובת הלוח מועברת באוגר r4 וממדי הלוח מועברים בשטח המשותף Width ו Height. פלט: 1 אם הלוח פתור, 0 אחרת. מוחזר במחסנית.
findInit	מקבלת את כתובת סדרת המהלכים, את אורכה ואת ממדי הלוח ומחזירה את הלוח שסדרת המהלכים פותרת. אם אין כזה השגרה תחזיר מערך מאותחל באפסים.	pc	קלט: כתובת סדרת המהלכים, אורכה והכתובת שלתוכה ייכתב הלוח מועברים במחסנית בסדר הנ"ל. ממדי הלוח מועברים בשטח המשותף Width ו Height. פלט: הלוח שנפתר ע"י ביצוע סדרת המהלכים ייכתב לכתובת היעד (המועברת בראש המחסנית).
nextLex	מקבלת מחרוזת המורכבת מהתווים {'D', 'L', 'R', 'U'} ואת אורכה ומחזירה את המחרוזת הבאה לפי סדר לקסיקוגרפי המורכבת מתווים אלו בלבד ובאותו האורך (ראו הסבר נוסף להלן).	pc	קלט: כתובת המחרוזת ואורכה מועברים במחסנית לפי הסדר הנ"ל. פלט: 1 במידה והתקבלה הסדרה האחרונה לפי סדר מילוני (כל התווים בה היו 'U') ו 0 אחרת. מוחזר ברגיסטר r4. שינוי המחרוזת בהתאם (באותה הכתובת).

שימו לב:

- i. אתם יכולים לממש כל שיגרת עזר בנוסף לשגרות שמפורטות למעלה.
- ii. כל אחת מהשגרות הנ"ל יכולה להשתמש בכל אחת מהשגרות האחרות.
- iii. למרות הדרישה לממש את כל השגרות למעלה, אין זה חובה להשתמש בשגרות אלה.

פירוט נוסף לגבי השיגרה nextLex:

שגרה זו מקבלת מחרוזת המורכבת מהתווים {'D', 'L', 'R', 'U'} בלבד ומחזירה את המחרוזת הבאה לפי סדר לקסיקוגרפי שהיא באותו האורך ומורכבת מאוסף תווים אלו. במידה והתקבלה הסדרה האחרונה לפי סדר מילוני (שכל התווים בה הם 'U') תוחזר הסדרה הראשונה בסדר (שבה כל התווים הם 'D') ויוחזר הערך 1 ברגיסטר r4. למשל, עבור מחרוזות באורך 2, אם נתחיל מהמחרוזת הראשונה בסדר DD ונקרא לפונקציה מספר פעמים נקבל:

מספר הקריאה		הערך שיוחזר ב r4
קריאה 1	L	D
קריאה 2	R	D
קריאה 3	U	D
קריאה 4	D	L
קריאה 5	L	L
קריאה 6	R	L
קריאה 7	U	L
קריאה 8	D	R
קריאה 9	L	R
.		
.		
.		
קריאה $2^4 - 1 = 15$	U	U
קריאה 16	D	D

תהליך בדיקת נכונות התוכנית

כחלק מבדיקת התרגיל, תיבדק גם נכונות הריצה של התוכנית. תהליך הבדיקה נעשה על ידי הוספת קלט (כלומר הוספת התוויות, ParLen, Board2, MovesTree, Width, Height, InitialBoard, Moves, MovesLen) לסוף הקובץ אותו אתם מגישים, וכן הוספת התוויות המשמשת לפלט (Output, Moves, MovesLen), כל אלו בכתובות מעל 8000. לכן, אין להשתמש בכתובות מעל 8000 בכתובת התוכנית. כמו כן, אין להגיש קובץ המכיל את הגדרות התוויות הנ"ל (שכן הגדרות אלו מוספות במהלך הבדיקה). אתם, כמובן, רשאים להוסיף תוויות אלו במהלך כתיבת התוכנית וניפוי השגיאות (debugging), אך, כאמור, אין להגיש את התוכנית שלכם עם הגדרת התוויות הנ"ל. לצורך הבהרת עניין זה, יסופקו שני קבצים: ex3_test.txt ו-ex3_test.bat. הקובץ ex3_test.txt מכיל את ההגדרות של תוויות אלו, והקובץ ex3_test.bat הוא קובץ הרצה המשמש להוספת התוויות. עליכם לבצע את הפעולות הבאות לפני הגשת התרגיל:

- יש לוודא כי שם הקובץ של התוכנית הוא ex3.s11,
- להוריד את שני הקבצים (ex3_test.txt ו-ex3_test.bat) מהאתר לאותו המיקום בו נמצא קובץ התוכנית.
- להריץ את הקובץ ex3_test.bat.
- ייצור קובץ חדש בשם ex3_temp.s11 המכיל את קוד התוכנית המקורי (מהקובץ ex3.s11) וכן את הגדרת התוויות (מהקובץ ex3_test.txt). יש לוודא כי עבור הקובץ החדש אין שגיאה בזמן תרגום וכי התוכנית מביאה לפלט הצפוי.
- בכל אופן, יש להגיש את הקובץ ex3.s11.

שימו לב: לא יתקבלו ערעורים הקשורים בעניין הטכני הנ"ל.

הערות נוספות

- על ה-main שלכם לבצע את שלוש המשימות.
- אפשר להניח שבמערך Moves מוקצים מספיק איברים להכלת כל סדרות המהלכים הנוצרות מהעץ.
- ניתן להניח שאורך סדרת המהלכים הארוכה ביותר יהיה לכל היותר 10 (בכל המשימות).
- ניתן להניח כי הלוח תקין (אין שני מספרים באותו אינדקס והאינדקס של כל מספר יהיה בין 1 ל $n \cdot m$).
- ניתן להניח שכל אחד ממימדי הלוח הוא לכל היותר 16.
- התוכנית צריכה לפעול נכון עבור כל קלט תקין.
- שימו לב לאותיות גדולות/קטנות בשימוש בכל התוויות.
- התוכנית צריכה לרוץ על הסימולטור המסופק באתר הקורס.
- יש להקפיד על תיעוד פנימי וחיצוני של התוכנית. יורדו נקודות בגין תיעוד לא מלא. ניתן לקרוא באתר הקורס ב-FAQ על רמת התיעוד הנדרשת.
- שאלות על התרגיל יש להפנות לאביב.
- הגשות באיחור יש לתאם לפני מועד ההגשה.
- הגשה לתא הקורס: תיעוד חיצוני מודפס בלבד (לא צריך להגיש את התוכנית מודפסת).
- הגשה אלקטרונית: קובץ הקוד ex3.s11 בלבד.
- ההגשה בזוגות בלבד!

עבודה נעימה!