

את"ם – תרגיל בית מס' 3 סמסטר חורף תשע"ז

תאריך פרסום: 13.12.2016 תאריך הגשה: 4.1.2017 (בשעה 23:55)
מתרגל אחראי על התרגיל: כפיר ניר-צבי (kfirnz@cs.technion.ac.il)

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

נושא התרגיל: שגרות + רקורסיה + 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 אלקטרונית דרך האתר (יש להגיש אלקטרונית רק את הקובץ ex3.s11. אין להגישו מכוון בתור קובץ 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.

לשאלה זו שני חלקים. חלק א' מתייחס אך ורק לשיגרה numX המופיעה בתוכנית הבאה, ללא תלות בשאר התוכנית. חלק ב' מתייחס לכל התוכנית.

1.	.=	torg	+1000	41.	numX:	mov	r0,	-(sp)
2.	main:	mov	pc, sp	42.		mov	4(sp),	-(sp)
3.		tst	-(sp)	43.		clr	10(sp)	
4.		tst	-(sp)	44.		clr	queue	
5.		tst	-(sp)	45.		mov	#8.,	r0
6.		mov	#list, -(sp)	46.	numXL:	mov	queue,	queue+2
7.		jsr	pc, findF	47.		mov	(sp),	queue
8.		tst	(sp)+	48.		bic	#177774,	queue
9.		mov	(sp)+, result	49.		cmp	queue,	queue+2
10.		mov	(sp)+, result+2	50.		blt	leave	
11.		halt		51.		inc	10(sp)	
12.	result:	.word	0, 0	52.		asr	(sp)	
13.	findF:	clr	4(sp)	53.		asr	(sp)	
14.		clr	6(sp)	54.		sob	r0,	numXL
15.		tst	@2(sp)	55.	leave:	tst	(sp)+	
16.		beq	quit	56.		mov	(sp)+,	r0
17.		tst	-(sp)	57.		rts	pc	
18.		tst	-(sp)	58.	queue:	.blkw	2	
19.		mov	6(sp), -(sp)					
20.		add	#2, (sp)	59.	list:	.word	20, 60, 10, 60, 200, 0	
21.		jsr	pc, findF					
22.		tst	-(sp)					
23.		mov	@12(sp), -(sp)					
24.		jsr	pc, numX					
25.		mov	@4(sp), -(sp)					
26.		jsr	pc, numX					
27.		tst	(sp)+					
28.		cmp	(sp)+, (sp)+					
29.		bhis	cont					
30.		clr	4(sp)					
31.	cont:	inc	4(sp)					
32.		tst	(sp)+					
33.		cmp	(sp), 2(sp)					
34.		bhis	setM					
35.		mov	2(sp), 10(sp)					
36.		br	skip1					
37.	setM:	mov	(sp), 10(sp)					
38.	skip1:	tst	(sp)+					
39.		mov	(sp)+, 6(sp)					
40.	quit:	rts	pc					

חלק א'

1. מלאו בטבלה הבאה (ברישום עשרוני) את הפלט שתחזיר השיגרה **numX** עבור כל אחד מהקלטים הבאים לשיגרה (ערכי הקלט רשומים בבסיס אוקטאלי).

הקלט ב 2(sp)	הפלט ב 4(sp)
4 ₈	
3220 ₈	
125224 ₈	

2. חיזרו על סעיף 1, **תחת ההנחה** שמחליפים את שורה 48 ב- **bic #177776, queue**.

הקלט ב 2(sp)	הפלט ב 4(sp)
4 ₈	
3220 ₈	
125224 ₈	

3. מהו הערך הגדול ביותר שיכול לקבל פרמטר הפלט של השיגרה **numX** עבור קלט כלשהו?

תשובה (בבסיס עשרוני):

4. מחליפים את שורה 45 ב- **mov #88., r0**. הביאו דוגמאות **לשלושה** ערכי קלט שונים שעבורם תחזיר השיגרה **numX** פלטים השונים מאלה שהייתה מחזירה (עבור אותם ערכי קלט) לפני השינוי הנ"ל.

תשובה (בבסיס אוקטאלי):

חלק ב'

5. עבור כל אחד מהפרמטרים אותם מקבלת השיגרה **findF**, 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 שתישארנה בטבלה שורות ריקות.

פרמטר מספר	שטח העברה	קלט / פלט / שניהם	ערך / כתובת
1			
2			
3			
4			
5			

6. תארו את תוכן המחסנית מיד לפני הביצוע הראשון של ההוראה בשורה 21. ניתן לכתוב ביטוי מהצורה "הכתובת של שורה 15" וניתן להשתמש בתוויות. הניחו שבתחילת התוכנית תוכן של אוגר i הוא i (עבור $i < 6$). אם קיים במחסנית ערך לא ידוע, כיתבו "לא ידוע" במקום המתאים.

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

7. מה יהיה הערך (ברישום עשרוני) של result בתום ריצת התוכנית? _____

8. מה יהיה הערך (ברישום עשרוני) של result+2 בתום ריצת התוכנית? _____

9. לפניכם מספר שינויים המוכנסים בשורה 59 בקוד (כל שינוי עומד בפני עצמו ואינו תלוי באחרים). לכל שינוי, כיתבו (ברישום עשרוני) מה יהיו ערכי המילים במערך result בתום ריצת התוכנית.

	שורה 59 הוחלפה ב-	result	result+2
א.	list: .word -1, 140, 1000, 0		
ב.	list: .word 7, 77, 0, 20, 200, 2000, 20000, 333		
ג.	list: .word 7, 77, 777, 7777, 0		

חלק רטוב (סולמות ונחשים)

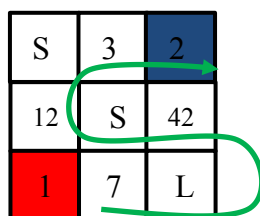
בתרגיל זה תתבקשו להרחיב את התוכנית שכתבתם בתרגיל בית 2, כלומר יהיה עליכם למצוא פתרון למשחק סולמות ונחשים (כמו בתרגיל בית 2) אך בנוסף תצרכו לחשב את סדרת המהלכים הקצרה ביותר שתוביל לסיום המשחק בהצלחה, ואת סדרת המהלכים בעלת סכום ניקוד מקסימלי.

תזכורת מתרגיל בית הקודם - תיאור כללי המשחק (לכל הפרטים תסתכלו בתיאור של תרגיל בית 2)

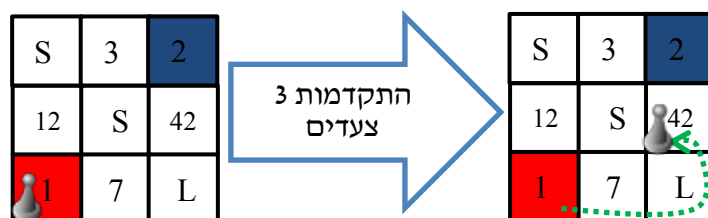
בכל מהלך, השחקן מטיל קובייה (סטנדרטית עם ערכים בין 1-6) ומתקדם כמספר הצעדים שמורה הקובייה.

1. במקרה ונחת על משבצת שבה יש **נחש** (מסומנת באות S) הוא **יורד** שורה בלוח (נשאר באותה עמודה) ומפעיל את כללים 1-3 בהתאם למשבצת שהגיע אליה.
2. במקרה ונחת על משבצת שבה יש **סולם** (מסומנת באות L) הוא **עולה** שורה בלוח (נשאר באותה עמודה) ומפעיל את כללים 1-3 בהתאם למשבצת שהגיע אליה.
3. במקרה שבו נחת על משבצת ממוספרת (משבצת עם מספר בין $0 - 50_8$) הוא סוכם את המספר לתוצאה הכללית שלו.

ההתקדמות בלוח היא בצורת **נחש** – מתחילים תמיד בהתקדמות ימינה. כשמגיעים למשבצת האחרונה בשורה ממשיכים למשבצת שמעליה בדיוק ו**מחליפים** את כיוון ההתקדמות באופן הבא –



לדוגמה, נתון הלוח ההתחלתי בשמאל בגודל 3×3 שבו משבצת ההתחלה מסומנת באדום (משבצת $[2,0]$) ומשבצת הסיום מסומנת בכחול (משבצת $[0,2]$). כאשר הקובייה תראה את המספר 3 נתקדם 3 צעדים, עד למשבצת $[1,2]$, כפי שניתן לראות באיור מימין (הפיון מייצג את השחקן).



מהלך ייחשב **לא** חוקי במקרים הבאים:

1. מהלך המסתיים במשבצת שמכילה סולם (L) כאשר המשבצת נמצאת בשורה העליונה ביותר.
2. מהלך המסתיים במשבצת שמכילה נחש (S) כאשר המשבצת נמצאת בשורה התחתונה ביותר.
3. מהלך שבמהלכו מדלגים מעל משבצת הסיום, ז"א ממשיכים להתקדם מעבר למשבצת הסיום.

סיום המשחק:

- המשחק מסתיים בהצלחה כאשר השחקן מגיע **בסיום המהלך האחרון למשבצת האחרונה**.
- המשחק מסתיים בכישלון אם בוצע מהלך לא חוקי (אחד משלושת המקרים שצוינו לעיל) או אם סדרת המהלכים לא מובילה את השחקן לנקודת הסיום.

הניקוד הכולל של השחקן הוא סכום המספרים במשבצות הממוספרות שבהן ביקר במהלך המשחק. כמו כן, יש לכלול גם משבצות ממוספרות שאליהן הגיע לאחר עליה בסולם או ירידה בנחש.

מספר הצעדים של השחקן הוא כמות המהלכים שהתבצעו.

מטרת התרגיל

בתרגיל זה תתבקשו לכתוב תכנית המקבלת לוח משחק ומחזירה סדרת מהלכים (כאמור, כל מהלך מציין מספר צעדים שיש להתקדם על הלוח), אשר מסתיימת במשבצת הסיום.

קלט התוכנית

- **nCols** – תווית המציינת מילה בזיכרון המכילה את מספר העמודות של הלוח. הניחו כי מספר העמודות הוא לכל היותר $2^7 - 1$ ולכל הפחות 1.
- **nRows** – תווית המציינת מילה בזיכרון המכילה את מספר השורות של הלוח. הניחו כי מספר השורות הוא לכל היותר $2^7 - 1$ ולכל הפחות 1.
- **Board** – תווית המציינת מערך של בתים המייצג את לוח המשחק. המערך יהיה בגודל $nCols \times nRows$ בתים, ויאוחסן בזיכרון שורה-שורה (כמו מערכים דו-ממדיים בשפת C). כל בית (משבצת בלוח) יכיל ערך בהתאם למפתח הבא:
 - מספר בין $0 - 50_8$ עבור משבצת שניתן לסכום לפתרון.
 - הערך ASCII של התו 'S' עבור משבצת שממנה יש לרדת משבצת אחת למטה (נחש).
 - הערך ASCII של התו 'L' עבור משבצת שממנה יש לעלות משבצת אחת למעלה (סולם).
- **numMoves** – מילה בזיכרון המכילה את האורך המקסימלי הנדרש של סדרת המהלכים הפותרת את הלוח.

פלט התוכנית

- **MovesShort** – תווית המציינת מערך של בתים אשר צריך להכיל את סדרת המהלכים הקצרה ביותר אשר פותרת את הלוח, סוף המערך יצוין ע"י ערך ה-ASCII של התו '@'.
 - **MovesScore** – תווית המציינת מערך של בתים אשר צריך להכיל את סדרת המהלכים אשר מניבה את הציון הגבוה ביותר אשר פותרת את הלוח, סוף המערך יצוין ע"י ערך ה-ASCII של התו '@'.
- במידה ולא ניתן לפתור את הלוח בתוך numMoves מהלכים או במידה שהלוח לא תקין, אז האיבר הראשון בכל אחד מהמערכים הללו צריך להכיל -1.

הערה

- אין להניח תקינות של הלוח (בניגוד לתרגיל בית 2) ועל כן יש לבדוק את תקינותו, כלומר:
1. התאים בלוח מכילים אך ורק מספרים בין $0 - 50_8$ או את הערך ה-ASCII של התווים S או L.
 2. שאין מצב שבו במשבצת מעל סולם יופיע נחש (למניעת לולאה אינסופית) או מצב שבו במשבצת הסיום יופיע נחש.

דוגמה לקלט אפשרי

```
nCols:      .word 5
nRows:      .word 3
numMoves:   .word 4

Board:      .byte  2, 3, 'S', 'L', 0
            .byte  5, 'L', 6, 'S', 6
            .byte  1, 'L', 'S', 1, 'L'
```

MovesShort: .blkw 10.

MovesScore: .blkw 10.

אז בסוף הריצה המערך MovesShort יכיל את הסדרה הבאה

1,3,'@

והמערך MovesScore יכיל את הסדרה הבאה

5,2,2,5,'@

דוגמה נוספת

```
nCols:      .word 5
nRows:      .word 3
numMoves:   .word 4

Board:      .byte  2, 3, 'S', 'L', 0
            .byte  5, 'L', 6, 'S', 6
            .byte  1, 'L', 'A', 1, 'L'
```

MovesShort: .blkw 10.

MovesScore: .blkw 10.

אז בסוף הריצה המערכים יכילו 1- בגלל שהלוח לא חוקי.

הערה:

במידה והלוח לא תקין או שאין פתרון אפשרי ב-numMoves מהלכים אז יש לכתוב 1- בתחילת מערכי הפלט.

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. שימו לב שיש **לתעד** כל אחת מהשגרות שאתם כותבים (גם כאלה שאינן ברשימה) כפי שמוסבר באתר הקורס.

שם השיגרה	תפקיד השיגרה	אוגר קישור	פרמטרים ושטח העברתם
checkBorad	מקבלת את הלוח ואת ממדי הלוח ומחזירה ובודקת האם הלוח תקין.	pc	קלט: הלוח מועבר בשטח המשותף Board ומימדי הלוח מועברים בשטח המשותף nCols ו nRows. פלט: תקינות הלוח תוחזר ברגיסטר r4. 0 במידה והלוח תקין ו-1 אחרת.
nextMoves	מקבלת מערך Moves ואורכו ומחזירה את סדרת המהלכים הבאה. (ראו הסבר מפורט בעמוד הבא).	r5	קלט: כתובת מערך המהלכים הנוכחי וגודלו מועברים inline בסדר הנתון (הכתובת מועברת ראשונה). פלט: 1 במידה והתקבלה הסדרה האחרונה לפי סדר מילוני (כל התווים בה היו 6) ו-0 אחרת. מוחזר ברגיסטר r4 בנוסף שינוי המחרוזת בהתאם (באותה הכתובת).
checkSol	מקבלת את הלוח, מימדי הלוח וסדרת מהלכים ובודקת האם סדרת המהלכים פותרת את הלוח. במידה וכן מחזירה את הניקוד שנצבר. שגרה זו מניחה שהלוח תקין.	pc	קלט: כתובת סדרת המהלכים מועברת באוגר r4. הלוח מועבר בשטח המשותף Board, וממדי הלוח מועברים בשטח המשותף nCols ו nRows. פלט: במידה וסדרת המהלכים מובילה לסיום בהצלחה אז מועבר הניקוד הכולל של סדרה זו. אחרת יוחזר -1. מועברים במחסנית לפי הסדר
findShortest	מקבלת את הלוח, ממדי הלוח, ואת מספר המהלכים המקסימלי המותר ומחזירה את סדרת המהלכים הקצרה ביותר הפותרת את הלוח. שגרה זו מניחה שהלוח תקין.	pc	קלט: הכתובת שלתוכה תכתב סדרת המהלכים וגודלה המקסימלי מועברות במחסנית. הלוח מועבר בשטח המשותף Board ומימדי הלוח מועברים בשטח המשותף nCols ו nRows. פלט: סדרת המהלכים הקצרה ביותר הפותרת את הלוח תכתב למערך היעד.
findScore	מקבלת את הלוח, ממדי הלוח, ואת מספר המהלכים המקסימלי המותר ומחזירה את סדרת אשר מניבה את הניקוד הגבוהה ביותר בתוך מספר המהלכים הנדרש. שגרה זו מניחה שהלוח תקין.	pc	קלט: הכתובת שלתוכה תכתב סדרת המהלכים וגודלה המקסימלי מועברות במחסנית. הלוח מועבר בשטח המשותף Board ומימדי הלוח מועברים בשטח המשותף nCols ו nRows. פלט: סדרת המהלכים בעלת ניקוד גבוהה ביותר הפותרת את הלוח תכתב למערך היעד.

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 `nextMoves` :

שגרה זו מקבלת סדרה המורכבת מהערכים {1,2,3,4,5,6} בלבד ומחזירה את הסדרה הבאה לפי סדר לקסיקוגרפי שהיא באותו האורך ומורכבת מאוסף הערכים אללו. במידה והתקבלה הסדרה האחרונה לפי סדר מילוני (שכל התווים בה הם 6) תוחזר הסדרה הראשונה בסדר (שבה כל התווים הם 1) ויוחזר הערך 1 ברגיסטר `r4`. למשל, עבור מחרוזות באורך 2, אם נתחיל מהמחרוזות הראשונה בסדר 11 ונקרא לפונקציה מספר פעמים נקבל :

מספר הקריאה	הערך שיוחזר ב <code>r4</code>	
קריאה 1	2	1
קריאה 2	3	1
קריאה 3	4	1
קריאה 4	5	1
קריאה 5	6	1
קריאה 6	1	2
קריאה 7	2	2
קריאה 8	3	2
קריאה 9	4	2
קריאה 10	5	2
קריאה 11	6	2
קריאה 12	1	3
.		
.		
.		
קריאה 35	6	6
קריאה 36	1	1

תהליך בדיקת נכונות התוכנית

כחלק מבדיקת התרגיל, תיבדק גם נכונות הריצה של התוכנית. תהליך הבדיקה נעשה על ידי הוספת הקלט (כלומר הוספת תוויות Board, numMoves, nCols, nRows) לסוף הקובץ אותו אתם מגישים, וכן הוספת תוויות המשמשות לפלט (התוויות MovesShort ו-MovesScore), כל אלו בכתובות מעל 5000. לכן, אין להשתמש בכתובות מעל 5000 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

- ניתן להניח כי **הקלט תקין**, כלומר:
 - גודל הלוח הוא nCols x nRows בתים.
 - במערכי הפלט מוקצים לפחות numMoves בתים להכלת סדרת המהלכים.
- התוכנית צריכה לפעול נכון עבור **כל** קלט תקין.
- שימו לב לאותיות **גדולות/קטנות** בשימוש בכל התוויות.
- אין לשנות** ערכים בלוח המשחק Board במהלך ריצת התוכנית (ייתכן ותחזרו אל תאים שכבר ביקרתם בהם).
- 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 **לכפיר ניר-צבי** בלבד.
- הגשות באיחור יש לתאם לפני מועד ההגשה.**
- הגשה לתא הקורס: הדפס התוכנית (הכולל בתוכו גם תיעוד פנימי) + תיעוד חיצוני.
- הגשה אלקטרונית: קובץ הקוד ex3.s11 בלבד.
- ההגשה בזוגות בלבד!**

עבודה נעימה!