

## את"ם – תרגיל בית מס' 3

### סמסטר חורף תשע"ח

תאריך פרסום: 12.12.2017 תאריך הגשה: 02.01.2018 (בשעה 23:55)  
מתרגל אחראי על התרגיל: שאדי אנדראוס (shadi.endrawis@gmail.com)

- הגשה בזוגת בלבד לתא ההגשה של הקורס ובאמצעות הגשה אלקטרונית.
- שאלות על התרגיל יש להפנות לשאדי אנדראוס shadi.endrawis@gmail.com.
- הגשות באיחור יש לתאם עם שאדי לפני מועד ההגשה הכללי.
- ערעורים והגשות באיחור יש להגיש לתא הדואר של שאדי בקומה 5.
- 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 שלו בצירוף תיעוד, כמפורט בהמשך, לתא הקורס. כדאי לקרוא באתר הקורס ב-FAQ על רמת התיעוד 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.

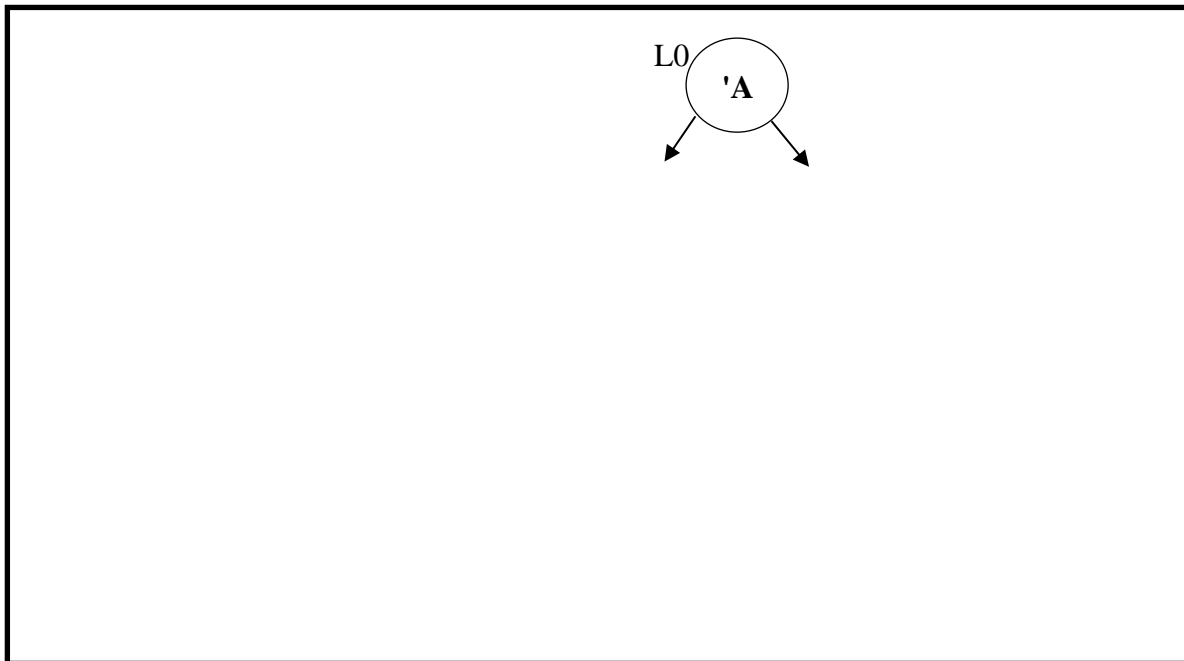
### חלק יבש

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

עיינו בתוכנית הבאה וענו על השאלות שאחריה.

1.	<b>NIL = 0</b>			46.	<b>chkT:</b>	mov	r0,	-(sp)
2.	.=torg	+ 1000		47.		mov	r1,	-(sp)
3.	<b>main:</b>	mov	pc,	48.		clr	r2	
4.		tst	-(sp)	49.		mov	10(sp),	-(sp)
5.		clr	r3	50.		mov	10(sp),	-(sp)
6.		mov	#10,	51.		jsr	pc,	chkN
7.	loop:	mov	#dict,	52.		tst	r2	
8.		sub	#2,	53.		bne	treeOut	
9.		add	r4,	54.		tst	12(sp)	
10.		add	r4,	55.		beq	treeOut	
11.		mov	@0(sp),	56.		tst	(sp)+	
12.		mov	#L0,	57.		mov	10(sp),	-(sp)
13.		jsr	pc,	58.		add	#2,	(sp)
14.		cmp	(sp)+,	59.		mov	@0(sp),	(sp)
15.		add	r2,	60.		jsr	pc,	chkT
16.		sob	r4,	61.		tst	r2	
17.		halt		62.		bne	treeOut	
18.				63.		mov	12(sp),	(sp)
19.	<b>chkN:</b>	mov	r0,	64.		add	#4,	(sp)
20.		mov	r1,	65.		mov	@0(sp),	(sp)
21.		clr	r2	66.		jsr	pc,	chkT
22.		cmpb	@10(sp),	67.	treeOut:	cmp	(sp)+,	(sp)+
23.		beq	success	68.		mov	(sp)+,	r1
24.		tst	6(sp)	69.		mov	(sp)+,	r0
25.		beq	out	70.		rts	pc	
26.		cmpb	@6(sp),	71.				
27.		bne	out	72.		L0: .word	'A, L5, L2	
28.		mov	10(sp),	73.		L1: .word	'A, NIL, L8	
29.		inc	(sp)	74.		L2: .word	'T, L4, L1	
30.		mov	10(sp),	75.		L3: .word	'A, NIL, NIL	
31.		add	#2,	76.		L4: .word	'E, L3, NIL	
32.		mov	@0(sp),	77.		L5: .word	'L, L7, NIL	
33.		jsr	pc,	78.		L6: .word	'V, L9, NIL	
34.		tst	r2	79.		L7: .word	'I, L6, NIL	
35.		bne	exit	80.		L8: .word	'M, NIL, NIL	
36.		mov	12(sp),	81.		L9: .word	'E, NIL, NIL	
37.		add	#4,	82.		dict: .word	text1, text2, text3, text4,	
38.		mov	@0(sp),	83.			text5, text6, text7, text8	
39.		jsr	pc,	84.		text1: .ascii	<MATAM@>	
40.	exit:	cmp	(sp)+,	85.		text2: .ascii	<TEA@>	
41.		br	out	86.		text3: .ascii	<ATE@>	
42.	success:	inc	r2	87.		text4: .ascii	<LIVE@>	
43.	out:	mov	(sp)+,	88.		text5: .ascii	<TEAM@>	
44.		mov	(sp)+,	89.		text6: .ascii	<VE@>	
45.		rts	pc	90.		text7: .ascii	<AE@>	
				91.		text8: .ascii	<AM@>	

1. בתוויות L0 עד L9 מועבר מידע במבנה של עץ. ציירו את העץ ששורשו מוגדר בתווית L0 ורשמו בכל צומת את תוכנו.



2. מלאו בטבלה הבאה את הפלט שתחזיר השיגרה **chkN** עבור כל אחד מהקלטים הבאים לשיגרה.

קלט ב 2(sp)	קלט ב 4(sp)	הפלט ב r2
#L0	#text3	
#L2	#text5	
#L0	#text8	

3. חזרו על סעיף 2, כאשר מחליפים את שורה 29 ב- **add #2, (sp)**.

קלט ב 2(sp)	קלט ב 4(sp)	הפלט ב r2
#L0	#text3	
#L2	#text5	
#L0	#text8	

4. חזרו על סעיף 2, כאשר מוחקים את שורה 41.

קלט ב 2(sp)	קלט ב 4(sp)	הפלט ב r2
#L0	#text3	
#L2	#text5	
#L0	#text8	

5. עבור כל אחד מהפרמטרים אותם מקבלת השיגרה **chkT**, רישמו בטבלה הבאה מהו שטח ההעברה שלו, ציינו אם הוא משמש לקלט, לפלט, או גם לפלט וגם לקלט, 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			

6. תארו את תוכן המחסנית מיד לפני הביצוע של ההוראה בשורה 19 בפעם **הראשונה**. 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, כתבו "לא ידוע" במקום המתאים.

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

7. מהם ערכי האוגרים  $r2$  ו- $r3$  בסוף ריצת התוכנית?

$r2$ : \_\_\_\_\_

$r3$ : \_\_\_\_\_

8. השלימו את השורות הבאות במקום שורות 82-91 כדי שבסוף ריצת התוכנית ערך האוגר r3 יהיה 3 וערך האוגר r2 יהיה 1.  
הקוד שמחליף את שורות 82-91:

```
82. dict: .word text1, text2, text3, text4, text5
83.      text6, text7, text8
84. text1: .ascii<_____>@>
85. text2: .ascii<_____>@>
86. text3: .ascii<_____>@>
87. text4: .ascii<_____>@>
88. text5: .ascii<_____>@>
89. text6: .ascii<VE @>
90. text7: .ascii<AE @>
91. text8: .ascii<AM @>
```

9. לפניכם מספר שינויים המוצעים בקוד (כל שינוי עומד בפני עצמו ואינו תלוי באחרים). עבור כל אחד מהשינויים, רשמו את ערכו של האוגר r3 בסוף ריצת התוכנית.

השינוי	ערך האוגר r3
א. שורה 12 הוחלפה בשורה <b>mov #L2, -(sp)</b>	
ב. שורה 13 הוחלפה בשורה <b>jsr pc, chkN</b>	
ג. שורה 6 הוחלפה בשורה <b>mov #4, r4</b>	

## חלק רטוב (הצפנת תמונה)

אליס ובוב גילו שהערבול שעשו בתרגיל הקודם היה פשוט מדי, כך שכל מי שקבל את התמונה המערבולת הצליח לשחזר אותה, כי פונקציות הערבול הייתה ידועה לכולם. לכן, רצו אליס ובוב להצפין את התמונות כך שאף אחד (מלבד אליס ובוב) לא יצליח לשחזר את התמונה. הם בחרו לעשות זאת בעזרת שימוש במפתח סודי שרק הם יודעים.

### תזכורת מתרגיל בית 2 - אלגוריתם ערבול תמונה

- הזזה ציקלית ימינה עבור שורה: החלפת כל איבר בשורה באיבר הקודם לו, כאשר האיבר הראשון (שאינו קודם) מוחלף באיבר האחרון.
- הזזה ציקלית שמאלה עבור שורה: החלפת כל איבר בשורה באיבר העוקב לו, כאשר האיבר האחרון (שאינו עוקב) מוחלף באיבר הראשון.
- פונקציית ערבול: פונקציה חז"ע ועל מקבוצת כל הערכים בין  $08-178$ .
- פונקציית שחזור: הפונקציה ההופכית של פונקציית הערבול.
- ערבול/שחזור בית: יתבצע ע"י הפעלת פונקציית הערבול/שחזור על ערך הבית.
- ערבול השורה ה-i בבלוק: יבוצע ע"י הפעלת הזזה ציקלית ימינה על השורה i פעמים, כאשר i הוא מספר השורה בבלוק (החל מ-0).
- שחזור השורה ה-i בבלוק: שחזור השורה ה-i בבלוק יבוצע ע"י הפעלת הזזה ציקלית שמאלה על השורה i פעמים, כאשר ספירת השורות מתחילה מ-0.
- ערבול בלוק: ערבול בלוק יבוצע ע"י ערבול כל בית בבלוק, ולאחר מכן ערבול כל שורה בבלוק.
- שחזור בלוק: שחזור בלוק פירושו "ביטול" של ערבול בלוק. כלומר, שחזור כל שורה בבלוק ולאחר מכן שחזור כל בית בבלוק.
- ערבול/שחזור תמונה: ערבול/שחזור כל הבלוקים בתמונה.

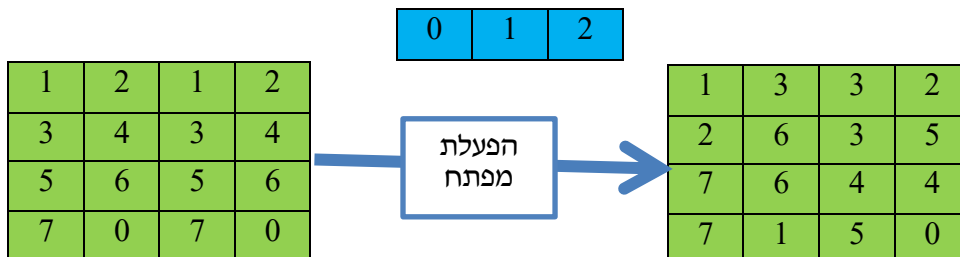
## אלגוריתם הצפנת ופענוח תמונות

כדי להצפין תמונות נשתמש במפתח סודי שרק אליס ובוב יודעים.

- מפתח: רשימה מקושרת, כאשר כל איבר ברשימה מכיל ערך בין  $0_8$  ו- $17_8$  ומצביע לאיבר הבא.
  - הפעלת מפתח על בלוק: ביצוע פעולת XOR בין האיבר ה- $i$  בבלוק לאיבר ה- $i$  במפתח, כאשר סדר המעבר על הבלוק הוא משמאל לימין, שורה אחרי שורה. אם גודל המפתח קטן מגודל הבלוק אז עושים XOR עם הערכים במפתח בצורה ציקלית (האיבר העוקב לאיבר האחרון במפתח הוא האיבר הראשון). דוגמה:
- נניח שהמפתח הוא באורך 3 והערכים שלו הם  $k_0, k_1, k_2$ , אזי עבור בלוק בגודל  $4 \times 4$  נבצע XOR של ערכי המפתח עם ערכי בלוק בצורה הבאה:

$k_0$	$k_1$	$k_2$	$k_0$
$k_1$	$k_2$	$k_0$	$k_1$
$k_2$	$k_0$	$k_1$	$k_2$
$k_0$	$k_1$	$k_2$	$k_0$

דוגמה להפעלת מפתח על בלוק:



- הפעלת מפתח על תמונה: הפעלת המפתח על כך בלוק בתמונה.
- הצפנת תמונה: תחבצע בשלבים הבאים:
  1. הפעלת המפתח הסודי על התמונה.
  2. ערבול התמונה הנוצרת משלב 1.
  3. הפעלת המפתח הסודי על התמונה משלב 2.
- פענוח תמונה: יתבצע בשלבים הבאים (שימו לב שהפעולה ההפוכה ל-XOR היא גם XOR):
  1. הפעלת המפתח הסודי על התמונה.
  2. שחזור התמונה משלב 1 (הפעולה ההפוכה מערבול).
  3. הפעלת המפתח הסודי על התמונה משלב 2.

## תיאור התרגיל

### חלק א

בדומה לתרגיל בית 2, עליכם לממש הצפנה של תמונה כפי שמוגדר למעלה. עליכם לקבל תמונה, פונקציית ערבול ומפתח, ולהחזיר תמונה מוצפנת בהתאם.

## חלק ב

אחרי ההצפנה המתוחכמת שתכננו אליס ובוב, מלורי (חברה של אליס ובוב) מצליחה להקשיב לשיחה בין אליס ובוב ברשת, ולקבל את התמונות שהם שולחים. למלורי אין את המפתח הסודי וכתוצאה מכך היא לא תצליח לפענח את התמונות המוצפנות שקיבלה. לרוע מזלם של אליס ובוב, יום אחד אליס שלחה את אותה תמונה פעמיים, פעם מוצפנת ופעם לא מוצפנת.

כעת יש למלורי את תמונה לפני ואחרי ההצפנה, ומזה היא רוצה לבנות מערכת למציאת המפתח הסודי. בתרגיל תקבלו שתי תמונות, אחת מוצפנת ואחת לא, ובנוסף תקבלו את פונקציית הערבול. עליכם לעבור על כל המפתחות האפשריים ולהחזיר את המפתח בעל האורך המינימאלי הגורם לפענוח/הצפנת התמונות.

## קלט התוכנית

התוכנית תקבל בתויות הבאות:

- nCols – תווית המציינת בית בזיכרון המכיל את מספר העמודות של התמונות. הניחו כי מספר העמודות הוא  $2^7-1$  לכל היותר.
- nRows – תווית המציינת בית בזיכרון המכיל את מספר השורות של התמונות. הניחו כי מספר השורות הוא  $2^7-1$  לכל היותר.
- OrigImg – תווית המציינת מערך בתים המייצג את התמונה הלא מוצפנת. המערך יהיה בגודל nRows x nCols בתים, ויאוחסן בזיכרון שורה-שורה (כמו מערכים דו-מימדיים בשפת C).
- EncImg – תווית המציינת מערך בתים המייצג את התמונה המוצפנת. המערך יהיה בגודל nRows x nCols בתים, ויאוחסן בזיכרון שורה-שורה (כמו מערכים דו-מימדיים בשפת C).
- Hash – תווית המציינת מערך בתים בזיכרון המייצג את פונקציית הערבול. הניחו כי גודלה המערך הוא בדיוק  $16_{10}$  בתים.
- Key – תווית המציינת רשימה מקושרת שכל איבר בה מכיל מצביע לאיבר הבא וערך בין  $0_8$  ל-  $17_8$  של איבר במפתח או 1- שמציין ערך לא חוקי שלא שייך למפתח. בתחילת התוכנית תינתן לכם רשימה שמכילה את כל המצביעים הנכונים וכל הערכים הם לא חוקיים, כלומר 1- (זה מייצג מפתח באורך 0).

**שימו לב:** הנחות לגבי תקינות הקלט מופיעות בסוף מסמך זה, תחת "הערות נוספות". אנא קראו אותן בעיון.

## פלט התוכנית

עליכם לדאוג כי המפתח בעל אורך מינימאלי הגורם לפענוח/הצפנת התמונות ייכתב לתוך הרשימה המקושרת המתחילה בתווית Key. במקרה בו תמונות הקלט לא מכילות ערכים תקנים מחזירים את המפתח ההתחלתי (כל הערכים שלו -1).



## שגרות

בתוכנית שלכם **עליכם** לממש מספר שגרות שיפורטו להלן. לכל שיגרה מוסבר תפקידה והמנשק שלה (מהו אוגר הקישור, כיצד היא מקבלת ומחזירה פרמטרים וכדומה). הקפידו לממש את המנשק **במדויק**. אסור לשגרות להסתמך על ערכו של אף משתנה גלובאלי אלא אם נאמר אחרת.

שם השיגרה	תפקיד השיגרה	אוגר קישור	פרמטרים ושטח העברתם
isLegal	מקבלת תמונה. מחזירה 1 אם התמונה חוקית ו-0 אחרת. תמונה חוקית היא תמונה שמכילה ערכים בין 0 ו-178.	r5	<b>קלט:</b> כתובת של התמונה, מספר השורות בתמונה ומספר העמודות בתמונה מועברים inline בסדר הנתון. <b>פלט:</b> 0 או 1 בהתאם אם התמונה חוקית או לא, יועבר כפלט ב- inline אחרי כל הקלטים.
encode	מקבלת תמונה לא מוצפנת, פונקציית ערבול ומפתח, ומחזירה תמונה מוצפנת לפי מפתח זה.	pc	<b>קלט:</b> כתובת של תמונה, מספר השורות בתמונה, מספר העמודות בתמונה, כתובת פונקציית הערבול וכתובת המפתח מועברים במחסנית בסדר הזה. <b>פלט:</b> התמונה תוצפן ותישמר במקומה של התמונה המקורית.
system	מקבלת שתי תמונות, אחת מוצפנת והשנייה לא. מחזירה את המפתח המתאים לפי מה שהוגדר בתרגיל. על השגרה לעבור על המפתחות בסדר שמוגדר ע"י שגרה nextKey.	pc	<b>קלט:</b> כתובת תמונה ראשונה, כתובת תמונה שנייה, מספר השורות של התמונות, מספר העמודות של התמונות וכתובת פונקציית הערבול מועברים במחסנית בסדר הזה. כתובת המפתח Key מועברת בשטח משותף. <b>פלט:</b> המפתח <b>הראשון</b> באורך מינימאלי הגורם להצפנת/פענוח התמונות במפתח Key.
cmpImg	מקבלת שתי תמונות. מחזירה 0 אם הן זהות ו-1 אם הן שונות.	pc	<b>קלט:</b> כתובת תמונה ראשונה, כתובת תמונה שנייה, מספר השורות בתמונות ומספר העמודות בתמונות מועברים במחסנית בסדר הזה (כלומר מספר העמודות נמצא בראש המחסנית). <b>פלט:</b> 0 או 1 בהתאם אם התמונות זהות או לא, מוחזר במחסנית (לפני הקלט).
nextKey	מקבלת מפתח ומחזירה את המפתח הבא בסדר מספרי (בצורה ציקלית). הסבר נוסף מופיע בעמוד הבא.	pc	<b>קלט:</b> כתובת המפתח מועברת באוגר r1. <b>פלט:</b> שינוי המפתח בהתאם.

## שימו לב:

- i. אתם יכולים לממש כל שיגרת עזר בנוסף לשגרות שמפורטות למעלה (למשל שגרת פענוח).
- ii. כל אחת מהשגרות הנ"ל יכולה להשתמש בכל אחת מהשגרות האחרות.

הסבר נוסף לגבי השגרה nextKey: שגרה זו מקבלת מצביע למפתח, ומחשבת את המפתח הבאה בתור לפי סדר המספרים.

השגרה מניחה שמתחילים ממפתח באורך 0 (כל הערכים ברשימה הם -1), לאחר  $16_{10}$  הפעלות של השגרה יש להגיע למפתח בגודל 1 שמכיל  $17_8$ , ולאחר  $(16^2+16)_{10}$  הפעלות של השגרה יש להגיע למפתח בגודל 2 שמכיל  $17_8$ -ים. כלומר, לאורך הדרך יש לעבור דרך כל המפתחות האפשריים בגודל 1 ואז דרך כל המפתחות האפשריים בגודל 2 וכך הלאה...

לדוגמה עבור רשימה נתונה בגודל שלוש:

$k_0$	$k_1$	$k_2$
-1	-1	-1

אז נקבל:

$k_0$	$k_1$	$k_2$	מספר הקריאה
0	-1	-1	קריאה 1
1	-1	-1	קריאה 2
2	-1	-1	קריאה 3
3	-1	-1	קריאה 4
			.
			.
			.
$17_8$	-1	-1	קריאה 16
0	0	-1	קריאה 17
1	0	-1	קריאה 18
			.
			.
			.
$17_8$	0	-1	קריאה 32
0	1	-1	קריאה 33
			.
			.
			.
$17_8$	$17_8$	-1	קריאה $16^2+16$
0	0	0	קריאה $16^2+16+1$
1	0	0	קריאה $16^2+16+2$
2	0	0	.
			.
			.
			.
$17_8$	$17_8$	$17_8$	קריאה $16^3+16^2+16$

## דוגמה לקלט ופלט

עבור הקלט הבא:

```
nCols: .byte 10
nRows: .byte 4
OrigImg: .byte 1, 2, 1, 2, 0, 1, 2, 3
          .byte 3, 4, 3, 4, 4, 5, 6, 7
          .byte 5, 6, 5, 6, 10, 11, 12, 13
          .byte 7, 0, 7, 0, 14, 15, 16, 17
EncImg:  .byte 3, 4, 3, 4, 2, 3, 4, 5
          .byte 7, 4, 7, 4, 10, 7, 6, 11
          .byte 7, 10, 7, 10, 14, 15, 12, 13
          .byte 3, 10, 3, 10, 16, 1, 0, 17
Hash:    .byte 2,3,4,5,6,7,10,11,12,13,14,15,16,17,0,1
Key:
node1:   .word -1, node2
node2:   .word -1, node3
node3:   .word -1, 0
```

התמונה המקורית (OrigImg):

1	2	1	2	0	1	2	3
3	4	3	4	4	5	6	7
5	6	5	6	10	11	12	13
7	0	7	0	14	15	16	17

בהתחלה המערכת תנסה מפתח בגודל 1 שערכו 0, ההצפנה שנקבל היא:

3	4	3	4	2	3	4	5
6	5	6	5	11	6	7	10
7	10	7	10	14	15	12	13
2	11	2	11	17	0	1	16

התמונה שהתקבלה לא זהה לתמונה המוצפנת שיש לנו, ולכן המערכת תמשיך לנסות מפתחות בסדר המוסבר למעלה עד שתגיע למפתח הנכון שהוא באורך 2, האיבר הראשון בו הוא 1 והשני 0, נקבל:

3	4	3	4	2	3	4	5
7	4	7	4	10	7	6	11
7	10	7	10	14	15	12	13
3	10	3	10	16	1	0	17

זו בדיוק התמונה שקיבלנו, ולכן זה המפתח שאנחנו מחפשים. לכן בסיום ריצת התוכנית, תוכן הרשימה המקושרת המתחילה בתווית Key :

```
Key:
node1: .word 1, node2
node2: .word 0, node3
node3: .word -1, 0
```

## תהליך בדיקת נכונות התוכנית

כחלק מבדיקת התרגיל, תיבדק גם נכונות הריצה של התוכנית. תהליך הבדיקה נעשה על ידי הוספת הקלט (כלומר הוספת התוויות Key, Hash, EncImg, OrigImg, nRows, nCols) לסוף הקובץ אותו אתם מגישים, כל אלו בכתובות מעל 11000. לכן, אין להשתמש בכתובות מעל 11000 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 הגדרות 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:

1. יש לוודא כי שם הקובץ של התוכנית הוא ex3.s11,
2. להוריד את שני הקבצים (ex3\_test.txt ו-ex3\_test.bat) מהאתר לאותו המיקום בו נמצא קובץ התוכנית.
3. להריץ את הקובץ ex3\_test.bat.
4. ייוצר קובץ חדש בשם ex3\_temp.s11 המכיל את קוד התוכנית המקורי (מהקובץ ex3.s11) וכן את הגדרות התוויות (מהקובץ ex3\_test.txt). יש לוודא כי עבור הקובץ החדש אין שגיאה בזמן תרגום וכי התוכנית מביאה לפלט הצפוי.
5. בכל אופן, יש להגיש את הקובץ ex3.s11.

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

## הערות נוספות

1. ניתן להניח כי **הקלט תקין**, כלומר:
  - התמונות ניתנות לחלוקה לבלוקים של 4x4.
  - המערך Hash מהווה פונ' חז"ע ועל מהתחום 08-17 לתחום 08-17.
  - הרשימה Key היא רשימה מקושרת תקינה שהאיבר האחרון בה מצביע ל-0.
  - מובטח שיש מפתח באורך קטן או שווה לאורך הרשימה הנתונה שמקיים את ההצפנה פענוח.
2. בתרגיל זה **אין להניח** כי ערכי המערים EncImg, OrigImg בטווח 08-17.
3. מפתח תקין תמיד מתחיל במספרים בין 0 ו-17, ומסתיים בערכים לא תקינים (-1). כלומר לא יתכן ערכים לא תקינים באמצע המפתח (ולא אמורים להגיע למקרה כזה).
4. התוכנית צריכה לפעול נכון עבור **כל** קלט תקין.
5. התוכנית צריכה לרוץ על הסימולטור המסופק באתר הקורס.
6. **יש להקפיד על תיעוד פנימי וחיצוני של התוכנית**. יורדו נקודות בגין תיעוד לא מלא. קיים מסמך באתר הקורס תחת לשונית תרגילי הבית המסביר **כיצד יש לתעד**. תיעוד חיצוני יהיה **לכל היותר** 4-5 עמודים, ולא צריך לכלול את הקוד שלכם.
7. שאלות על התרגיל יש להפנות **לשאדי אנדראוס** בלבד.
8. **הגשות באיחור יש לתאם לפני מועד ההגשה**.
9. **הגשה לתא הקורס**: דף שער (נמצא באתר הקורס) + תשובות לחלק היבש (ללא דפי ההוראות של החלק הרטוב) + תיעוד חיצוני (**לכל היותר** 4-5 עמודים).
10. **ההגשה בזוגות בלבד!**

עבודה נעימה!