

דיסקליימר: הקובץ הוא תוצאה של עבודה פרטית שלי על סיכום ההנחיות וליקוט ההבדלים בין הדרישות הנוכחיות לדרישות של הסמסטר הקודם.

היעזרו בו אבל אל תסתמכו עליו ב-100%. אם נפלה טעות בקובץ אני מתנצל מראש אך האחריות לוודא תקינות של ההנחיות היא עליכם לגמרי (ובינינו, מה זה דורש? רפרוף קל על החוברת במקביל לקובץ?)

מוזמנים כמובן להתריע בקבוצות ולעדכן כדי שהבאים בתור יזהרו יותר...

יאללה, שיהיה לנו בהצלחה!

עדכון:

בשאלות מהפורום הוספתי באדום הערות של אסתר (המנחה) שמאשרות סופית מה נדרש בשאלות בהן לא הייתה הכרעה ברורה. אסתר עברה על התוכן של העמודה הימנית בקובץ וענתה לבסוף על שאלות.

מאפייני מחשב	סמסטר נוכחי	סמסטר עבר (2023)																																																													
מבנה הוראת מכונה	מילה אחת עד 3 בגג. מילה ראשונה היא הפקודה, השאר מייצגות אופרנדים. מבנה מילה ראשונה זהה עבור כל סוג פקודה. מבנה בינארי: <table><tr><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="3">מיעון אופרנד מקור</td><td colspan="3">Opcode</td><td colspan="3">מיעון אופרנד יעד</td><td colspan="3">A,R,E</td></tr></table>	11	10	9	8	7	6	5	4	3	2	1	0	מיעון אופרנד מקור			Opcode			מיעון אופרנד יעד			A,R,E			מעבד עם רגיסטר PSW המכיל מספר דגלים המאפיינים את מצב הפעילות במעבד בכל רגע נתון. אוגרים (r0-7) פריפיקס: @ זיכרון - 1024 בתים (0-1023) קידוד ההוראה הראשונה מתחיל במען 100 (כלומר 0-99 תפוסים). 12 סיביות (0-11) מספרים שלמים חיוביים ושליילים בלבד, ללא תמיכה בממשיים. 2's complement תווים בקוד ascii																																					
	11	10	9	8	7	6	5	4	3	2	1	0																																																			
מיעון אופרנד מקור			Opcode			מיעון אופרנד יעד			A,R,E																																																						
מילה אחת עד 4 בגג. מילה ראשונה היא הפקודה, השאר מייצגות אופרנדים. מבנה מילה ראשונה זהה עבור כל סוג פקודה. מבנה בינארי: <table><tr><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="4">פרמטר 1</td><td colspan="4">פרמטר 2</td><td colspan="4">opcode</td><td colspan="2">מיעון אופרנד מקור</td></tr></table>	13	12	11	10	9	8	7	6	5	4	3	2	1	0	פרמטר 1				פרמטר 2				opcode				מיעון אופרנד מקור		קידוד המילה בסוף ייעשה בבסיס 2 ייחודי: "/" = 1, "." = 0 opcode זה הקוד המייצג את הפקודות הקיימות, אלו הפקודות: <table><tr><th>שם הפעולה</th><th>קוד הפעולה בבסיס דצימלי (10)</th></tr><tr><td>mov</td><td>0</td></tr><tr><td>cmp</td><td>1</td></tr><tr><td>add</td><td>2</td></tr><tr><td>sub</td><td>3</td></tr><tr><td>not</td><td>4</td></tr><tr><td>clr</td><td>5</td></tr><tr><td>lea</td><td>6</td></tr><tr><td>inc</td><td>7</td></tr><tr><td>dec</td><td>8</td></tr><tr><td>jmp</td><td>9</td></tr><tr><td>bne</td><td>10</td></tr><tr><td>red</td><td>11</td></tr><tr><td>prn</td><td>12</td></tr><tr><td>jsr</td><td>13</td></tr><tr><td>rts</td><td>14</td></tr><tr><td>stop</td><td>15</td></tr></table>	שם הפעולה	קוד הפעולה בבסיס דצימלי (10)	mov	0	cmp	1	add	2	sub	3	not	4	clr	5	lea	6	inc	7	dec	8	jmp	9	bne	10	red	11	prn	12	jsr	13	rts	14	stop	15
13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																		
פרמטר 1				פרמטר 2				opcode				מיעון אופרנד מקור																																																			
שם הפעולה	קוד הפעולה בבסיס דצימלי (10)																																																														
mov	0																																																														
cmp	1																																																														
add	2																																																														
sub	3																																																														
not	4																																																														
clr	5																																																														
lea	6																																																														
inc	7																																																														
dec	8																																																														
jmp	9																																																														
bne	10																																																														
red	11																																																														
prn	12																																																														
jsr	13																																																														
rts	14																																																														
stop	15																																																														

שיטות מיעון:

קוד	שיטת מיעון	תוכן מילים נוספות	אופן כתיבה
1	מיידי Imm	מילה נוספת, נתון, 10 סיביות + ARE	מספר שלם בעשרוני ללא פריפיקס
3	ישר Directive	מילה נוספת, כתובת, 10 סיביות + ARE	תווית (שהוצהרה או תוצהר)
5	אוגר ישר Register	מילה נוספת, סיביות 2-6 אוגר יעד, 7-11 אוגר מקור + ARE	שם של אוגר, עם פריפיקס @

סיביות אלה מתווספות רק לקידודים של הוראות (לא של נתונים), והן מתווספות גם לכל המילים הנוספות שיש לקידודים אלה.

שיטות מיעון:

קוד	שיטת מיעון	תוכן מילים נוספות	אופן כתיבה
0	מיידי Imm	מילה נוספת, נתון, 12 סיביות + ARE	מספר שלם בעשרוני עם פריפיקס #
1	ישר Directive	מילה נוספת, כתובת, 12 סיביות + ARE	תווית (שהוצהרה או תוצהר)
2	קפיצה עם פרמטרים label w param	רק לפעולות קפיצה. העברת פרמטרים לקפיצה. עד 3 מילים נוספות, מורכב...	תווית עם סוגריים ופרמטרים
3	אוגר ישר Register	מילה נוספת, סיביות 2-7 אוגר יעד, 8- 13 אוגר מקור + ARE	שם של אוגר, ללא פריפיקס

מפרט הוראות
מכונה

PC - רגיסטר מעבד (לא רגיסטר כללי) שמכיל בכל רגע נתון את הכתובת של מילת הזיכרון שמתבצעת (המילה הראשונה).

- רגיסטר מעבד (לא רגיסטר כללי) שמכיל בכל רגע נתון את הכתובת של מילת הזיכרון הנוכחית שמתבצעת (המילה הראשונה).

קבוצות הוראות המכונה:

קבוצות הוראות המכונה:

1. פקודות בעלות 2 אופרנדים:

הוראה	הפעולה המתבצעת	דוגמה	הסבר הדוגמה
mov	מבצעת העתקה של האופרנד הראשון, אופרנד המקור (source) אל האופרנד השני, אופרנד היעד (destination) (בהתאם לשיטת המיעון).	mov A, @r1	העתק את תוכן המשתנה A (המילה שבכתובת A בזכרון) אל רגיסטר r1.
cmp	מבצעת "השוואה" בין שני האופרנדים שלה. אופן ההשוואה: תוכן אופרנד היעד (השני) מופחת מתוכן אופרנד המקור (הראשון), ללא שמירת תוצאת החיסור. פעולת החיסור מעדכנת דגל בשם Z ("דגל האפס") ברגיסטר הסטטוס (PSW).	cmp A, @r1	אם תוכן המשתנה A זהה לתוכנו של רגיסטר r1 אזי דגל האפס, Z, ברגיסטר הסטטוס (PSW) יודלק, אחרת הדגל יאופס.
add	אופרנד היעד (השני) מקבל את תוצאת החיבור של אופרנד המקור (הראשון) והיעד (השני).	add A, @r0	רגיסטר r0 מקבל את תוצאת החיבור של תוכן המשתנה A ותוכנו הנוכחי של רגיסטר r0.
sub	אופרנד היעד (השני) מקבל את תוצאת החיסור של אופרנד המקור (הראשון) מאופרנד היעד (השני).	sub 3, @r1	רגיסטר r1 מקבל את תוצאת החיסור של הערך 3 מתוכנו הנוכחי של רגיסטר r1.
lea	lea הוא קיצור (ראשי תיבות) של: load effective address. פעולה זו מציבה את המען בזיכרון המיוצג על ידי התווית (המקור), שבאופרנד הראשון (המקור), אל אופרנד היעד (האופרנד השני).	lea HELLO, @r1	המען שמייצגת התווית HELLO מוצב לרגיסטר r1.

2. פקודות בעלות אופרנד בודד:

הוראה	הפעולה המתבצעת	דוגמה	הסבר דוגמה
not	היפוך ערכי הסיביות באופרנד (כל סיבית שערכה 0 תהפוך ל-1 ולהיפך: 1 ל-0).	not @r2	r2 ← not r2
clr	איפוס תוכן האופרנד.	clr @r2	r2 ← 0
inc	הגדלת תוכן האופרנד באחד.	inc @r2	r2 ← r2 + 1
dec	הקטנת תוכן האופרנד באחד.	dec C	C ← C - 1

2. פקודות בעלות אופרנד בודד:

הוראה	הפעולה המתבצעת	דוגמה	הסבר הדוגמה
mov	מבצעת העתקה של האופרנד הראשון, אופרנד המקור (source) אל האופרנד השני, אופרנד היעד (destination) (בהתאם לשיטת המיעון).	mov A, r1	העתק תוכן המשתנה A (המילה שבכתובת A בזכרון) לרגיסטר r1.
cmp	מבצעת "השוואה" בין שני האופרנדים שלה. אופן ההשוואה: תוכן אופרנד היעד (השני) מופחת מתוכן אופרנד המקור (הראשון), ללא שמירת תוצאת החיסור. פעולת החיסור מעדכנת דגל בשם Z ("דגל האפס") ברגיסטר הסטטוס (PSW).	cmp A, r1	אם תוכן המשתנה A זהה לתוכנו של רגיסטר r1 אזי דגל האפס, Z, ברגיסטר הסטטוס (PSW) יודלק, אחרת הדגל יאופס.
add	אופרנד היעד (השני) מקבל את תוצאת החיבור של אופרנד המקור (הראשון) והיעד (השני).	add A, r0	רגיסטר r0 מקבל את תוצאת החיבור של תוכן המשתנה A ותוכנו הנוכחי של רגיסטר r0.
sub	אופרנד היעד (השני) מקבל את תוצאת החיסור של אופרנד המקור (הראשון) מאופרנד היעד (השני).	sub #3, r1	רגיסטר r1 מקבל את תוצאת החיסור של הערך 3 מתוכנו הנוכחי של רגיסטר r1.
lea	lea הינו ראשי תיבות של: load effective address. פעולה זו מבצעת טעינה של המען בזיכרון המיוצג על ידי התווית (המקור), שבאופרנד הראשון (המקור), אל אופרנד היעד (האופרנד השני).	lea HELLO, r1	המען שמייצגת התווית HELLO מוכנס לרגיסטר r1.

jmp	קפיצה (הסתעפות) בלתי מותנית אל ההוראה שנמצאת במען המיוצג על ידי האופרנד. כלומר, בעת ביצוע ההוראה, מצביע התוכנית (PC) יקבל את ערך האופרנד היעד.	jmp LINE	PC ← LINE מצביע התוכנית מקבל את המען המיוצג על ידי התווית LINE, ולפיכך הפקודה הבאה שתבצע תהיה במען זה.
bne	bne הוא קיצור (ראשי תיבות) של branch if not equal (to zero). זוהי הוראה הסתעפות מותנית. מצביע התוכנית (PC) יקבל את ערך האופרנד היעד אם ערכו של הדגל Z באוגר הסטטוס (PSW) הינו 0. כזכור, ערך הדגל Z נקבע בהוראות cmp.	bne LINE	אם ערך הדגל Z ברגיסטר הסטטוס (PSW) הינו 0: PC ← LINE
red	קריאה של תו מהקלט הסטנדרטי (stdin) אל האופרנד.	red @r1	קוד ה-ascii של התו הנקרא מהקלט הסטנדרטי ייכנס לאוגר r1.
prn	הדפסת התו הנמצא באופרנד, הפלט הסטנדרטי (stdout).	prn @r1	התו אשר קוד ה-ascii שלו נמצא באוגר r1 יודפס לפלט הסטנדרטי.
jsr	קריאה לשגרה (סברוטין), מצביע התוכנית (PC) הנוכחי נדחף לתוך המחשנית שבוכרן המחשב, והאופרנד מוכנס ל-PC.	jsr FUNC	push(PC) PC ← FUNC

סיביות 9-11 יכילו תמיד 0 כי אין אופרנד מקור אלא רק יעד.

3. פקודות ללא אופרנד:

הוראה	הפעולה המתבצעת	דוגמה	הסבר דוגמה
rts	חזרה משיגרה. הערך בראש המחשנית של זמן-ריצה מוצא מן המחשנית ומוכנס אל מצביע התוכנית (PC).	rts	PC ← pop()
stop	עצירת ריצת התוכנית.	stop	התכנית עוצרת

בעלות מילה אחת בלבד (המילה הראשונה).
מיעון האופרנדים: 0.

הוראה	הפעולה המתבצעת	דוגמה	הסבר דוגמה
not	היפוך ערכי הסיביות באופרנד (כל סיבית שערכה 0 תהפוך ל-1 ולהפך: 1 ל-0).	not r2	r2 ← not r2
clr	איפוס תוכן האופרנד.	clr r2	r2 ← 0
inc	הגדלת תוכן האופרנד באחד.	inc r2	r2 ← r2 + 1
dec	הקטנת תוכן האופרנד באחד.	dec C	C ← C - 1
jmp	קפיצה בלתי מותנית (עם או בלי פרמטרים) אל ההוראה שנמצאת במען המיוצג על ידי האופרנד. כלומר, בעת ביצוע ההוראה, מצביע התוכנית (PC) יקבל את ערך האופרנד היעד.	jmp LINE או jmp LINE(#-6,r4)	PC ← LINE r6 ← -6 r7 ← r4
bne	bne הינו ראשי תיבות של: branch if not equal (to zero). זוהי הוראה הסתעפות מותנית (עם או בלי פרמטרים). מצביע התוכנית (PC) יקבל את ערך האופרנד היעד אם ערכו של הדגל Z ברגיסטר הסטטוס (PSW) הינו 0. כזכור, הדגל Z נקבע בהוראות cmp.	bne LINE או bne LINE(X,r4)	אם ערך הדגל Z ברגיסטר הסטטוס (PSW) הינו 0: PC ← LINE אם ערך הדגל Z ברגיסטר הסטטוס (PSW) הינו 0: PC ← LINE r6 ← word at X r7 ← r4
red	קריאה של תו מהקלט הסטנדרטי (stdin) אל האופרנד.	red r1	קוד ה-ascii של התו הנקרא מהקלט הסטנדרטי ייכנס לאוגר r1.
prn	הדפסת התו הנמצא באופרנד, הפלט הסטנדרטי (stdout).	prn r1	התו אשר קוד ה-ascii שלו נמצא ברגיסטר r1 יודפס לפלט הסטנדרטי.
jsr	קריאה לשגרה (סברוטין), עם או בלי פרמטרים. מצביע התוכנית (PC) הנוכחי נדחף לתוך המחשנית שבוכרן המחשב, והאופרנד מוכנס ל-PC.	jsr FUNC או jsr FUNC(#75,X)	push(PC) PC ← FUNC r6 ← 75 r7 ← word at X

סיביות 4-5 יכילו תמיד 0 כי אין אופרנד מקור אלא רק יעד.

3. פקודות ללא אופרנד:

הוראה	הפעולה המתבצעת	דוגמה	הסבר דוגמה
rts	חזרה משיגרה. הערך בראש המחשנית של זמן-ריצה מוצא מן המחשנית ומוכנס אל מצביע התוכנית (PC).	rts	ההוראה הבאה שתבצע תהיה זו שאחרי ההוראה jsr שקראה לשגרה.
stop	עצירת ריצת התוכנית.	stop	התכנית עוצרת מיידית

מיעון האופרנדים: 0.

מאקרו: מתחיל במילה mcro ובשם המאקרו מסתיים במילה endmcro	מאקרו: מתחיל במילה mcro ובשם המאקרו מסתיים במילה endmcro
שימוש במאקרו - אזכור שם המאקרו	שימוש במאקרו - אזכור שם המאקרו
התוכנית לאחר פרישת המקרו היא התוכנית שהאסמבלר אמור לתרגם.	התוכנית לאחר פרישת המקרו היא התוכנית שהאסמבלר אמור לתרגם.
<ul style="list-style-type: none"> אין במערכת הגדרות מאקרו מקוננות. שם של הוראה או הנחיה לא יכול להיות שם של מאקרו. ניתן להניח שלכל שורת מאקרו בקוד המקור קיימת סגירה עם שורת endmcro (אין צורך לבדוק זאת) הגדרת מאקרו תהיה תמיד לפני הקריאה למאקרו נדרש שהקדם-אסמבלר ייצור קובץ עם הקוד המורחב הכולל פרישה של המאקרו (הרחבה של קובץ המקור המתואר בהמשך). "קובץ המקור המורחב" הוא "קובץ מקור" לאחר פרישת המאקרו, לעומת "קובץ מקור ראשוני" שהוא קובץ הקלט למערכת, כולל הגדרת המאקרואים. 	<ul style="list-style-type: none"> אין במערכת הגדרות מאקרו מקוננות. שם של הוראה או הנחיה לא יכול להיות שם של מאקרו. ניתן להניח שלכל שורת מאקרו בקוד המקור קיימת סגירה עם שורת endmcro (אין צורך לבדוק זאת) הגדרת מאקרו תהיה תמיד לפני הקריאה למאקרו נדרש שהקדם-אסמבלר ייצור קובץ עם הקוד המורחב הכולל פרישה של המאקרו (הרחבה של קובץ המקור המתואר בהמשך). "קובץ המקור המורחב" הוא "קובץ מקור" לאחר פרישת המאקרו, לעומת "קובץ מקור ראשוני" שהוא קובץ הקלט למערכת, כולל הגדרת המאקרואים.
משפטים: קובץ מקור בשפת אסמבלי מורכב משורות המכילות משפטים של השפה, כאשר כל משפט מופיע בשורה נפרדת. כלומר, ההפרדה בין משפט למשפט בקובץ המקור הינה באמצעות התו (') \שורה חדשה).	משפטים: קובץ מקור בשפת אסמבלי מורכב משורות המכילות משפטים של השפה, כאשר כל משפט מופיע בשורה נפרדת. כלומר, ההפרדה בין משפט למשפט בקובץ המקור הינה באמצעות התו (') \שורה חדשה).
אורך שורה בקובץ המקור: 80 תווים לכל היותר (לא כולל התו \n).	אורך שורה בקובץ המקור: 80 תווים לכל היותר (לא כולל התו \n).
4 סוגי משפטים:	4 סוגי משפטים:

מבנה תכנית בשפת אסמבלי	מבנה תכנית בשפת אסמבלי
מאקרו: מתחיל במילה mcro ובשם המאקרו מסתיים במילה endmcro	מאקרו: מתחיל במילה mcro ובשם המאקרו מסתיים במילה endmcro
שימוש במאקרו - אזכור שם המאקרו	שימוש במאקרו - אזכור שם המאקרו
התוכנית לאחר פרישת המקרו היא התוכנית שהאסמבלר אמור לתרגם.	התוכנית לאחר פרישת המקרו היא התוכנית שהאסמבלר אמור לתרגם.
<ul style="list-style-type: none"> אין במערכת הגדרות מאקרו מקוננות. שם של הוראה או הנחיה לא יכול להיות שם של מאקרו. ניתן להניח שלכל שורת מאקרו בקוד המקור קיימת סגירה עם שורת endmcro (אין צורך לבדוק זאת) הגדרת מאקרו תהיה תמיד לפני הקריאה למאקרו נדרש שהקדם-אסמבלר ייצור קובץ עם הקוד המורחב הכולל פרישה של המאקרו (הרחבה של קובץ המקור המתואר בהמשך). "קובץ המקור המורחב" הוא "קובץ מקור" לאחר פרישת המאקרו, לעומת "קובץ מקור ראשוני" שהוא קובץ הקלט למערכת, כולל הגדרת המאקרואים. 	<ul style="list-style-type: none"> אין במערכת הגדרות מאקרו מקוננות. שם של הוראה או הנחיה לא יכול להיות שם של מאקרו. ניתן להניח שלכל שורת מאקרו בקוד המקור קיימת סגירה עם שורת endmcro (אין צורך לבדוק זאת) הגדרת מאקרו תהיה תמיד לפני הקריאה למאקרו נדרש שהקדם-אסמבלר ייצור קובץ עם הקוד המורחב הכולל פרישה של המאקרו (הרחבה של קובץ המקור המתואר בהמשך). "קובץ המקור המורחב" הוא "קובץ מקור" לאחר פרישת המאקרו, לעומת "קובץ מקור ראשוני" שהוא קובץ הקלט למערכת, כולל הגדרת המאקרואים.
משפטים: קובץ מקור בשפת אסמבלי מורכב משורות המכילות משפטים של השפה, כאשר כל משפט מופיע בשורה נפרדת. כלומר, ההפרדה בין משפט למשפט בקובץ המקור הינה באמצעות התו (') \שורה חדשה).	משפטים: קובץ מקור בשפת אסמבלי מורכב משורות המכילות משפטים של השפה, כאשר כל משפט מופיע בשורה נפרדת. כלומר, ההפרדה בין משפט למשפט בקובץ המקור הינה באמצעות התו (') \שורה חדשה).
אורך שורה בקובץ המקור: 80 תווים לכל היותר (לא כולל התו \n).	אורך שורה בקובץ המקור: 80 תווים לכל היותר (לא כולל התו \n).
4 סוגי משפטים:	4 סוגי משפטים:

סוג המשפט	הסבר כללי
משפט ריק	זוהי שורה המכילה אך ורק תווים לבנים (whitespace). כלומר מכילה רק את התווים ' ' ו- ' ' (טאבים ורווחים). ייתכן ובשורה אין אף תו (למעט התו null, כלומר השורה ריקה).
משפט הערה	זוהי שורה בה התו הראשון הוא ' '; (נקודה פסיק). על האסמבלר להתעלם לחלוטין משורה זו.
משפט הנחיה	זהו משפט המנחה את האסמבלר מה עליו לעשות כשהוא פועל על תכנית המקור. יש מספר סוגים של משפטי הנחיה. משפט הנחיה עשוי לגרום להקצאת זיכרון ואתחול משתנים של התכנית, אך הוא אינו מייצר קידוד של הוראות מכונה המיועדות לביצוע בעת ריצת התוכנית.
משפט הוראה	זהו משפט המייצר קידוד של הוראות מכונה לביצוע בעת ריצת התוכנית. המשפט מורכב משם ההוראה (פעולה) שעל המעבד לבצע, והאופרנדים של ההוראה.
משפט הנחיה:	
תווית (אופציונלי), שם ההנחיה (מתחיל ב-. וכולו אותיות קטנות), פרמטרים.	
ללא ARE, תופס את כל 12 הסיביות של המילה.	
1. data - מספרים שלמים חוקיים מופרדים ע"י פסיק. רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.	1. data - מספרים שלמים חוקיים מופרדים ע"י פסיק. רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.
ללא פסיק לפני או אחרי אוסף הפרמטרים.	ללא פסיק לפני או אחרי אוסף הפרמטרים.
משמש להקצאת תאי זיכרון לאחסון הנתונים, לאחריו מתבצע קידום מונה הנתונים בהתאם למספר הערכים.	משמש להקצאת תאי זיכרון לאחסון הנתונים, לאחריו מתבצע קידום מונה הנתונים בהתאם למספר הערכים.
אם ישנה תווית אז היא נכנסת אל טבלת הסמלים עם ערך מונה הנתונים לפני הקידום (וזוהו בעצם מיקום הנתון הראשון בסט שהתווסף).	אם ישנה תווית אז היא נכנסת אל טבלת הסמלים עם ערך מונה הנתונים לפני הקידום (וזוהו בעצם מיקום הנתון הראשון בסט שהתווסף).
2. string - פרמטר בודד - מחזרות ascii עם מרכאות. כל תו הופך למילה בפני עצמה.	2. string - פרמטר בודד - מחזרות ascii עם מרכאות. כל תו הופך למילה בפני עצמה.
בסוף מתווסף הערך 0 (\0) שמסמן את סוף המחזרות.	בסוף מתווסף הערך 0 (\0) שמסמן את סוף המחזרות.
תווית - בדומה למה שקורה בדאטה, מסמלת את כתובת תחילת המחזרות.	תווית - בדומה למה שקורה בדאטה, מסמלת את כתובת תחילת המחזרות.
3. entry - פרמטר אחד - שם תווית שמקבלת את ערכה בקובץ הנוכחי ומאפשרת לקבצים אחרים להשתמש בה כאופרנד של הוראה.	3. entry - פרמטר אחד - שם תווית שמקבלת את ערכה בקובץ הנוכחי ומאפשרת לקבצים אחרים להשתמש בה כאופרנד של הוראה.
לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.	לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.
4. extern - פרמטר אחד - שם של תווית שמוגדרת בקובץ אחר ונרצה להשתמש בה פה. הכיוון ההפוך של entry. לא ניתן להגדיר את 2 ההנחיות האלה על אותה תווית באותו קובץ.	4. extern - פרמטר אחד - שם של תווית שמוגדרת בקובץ אחר ונרצה להשתמש בה פה. הכיוון ההפוך של entry. לא ניתן להגדיר את 2 ההנחיות האלה על אותה תווית באותו קובץ.
לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.	לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.
משפט הוראה	
מורכב מ:	
1. תווית (אופציונלי) - אם קיימת תוכנס לטבלת הסמלים וערכה יהיה מען המילה הראשונה של ההוראה בתוך הקוד שבונה האסמבלר	1. תווית (אופציונלי) - אם קיימת תוכנס לטבלת הסמלים וערכה יהיה מען המילה הראשונה של ההוראה בתוך הקוד שבונה האסמבלר
2. שם הפעולה	2. שם הפעולה
3. אופרנדים אם אמורים להיות.	3. אופרנדים אם אמורים להיות.
שם הפעולה תמיד באותיות קטנות, והוא אחת מ- 16 הפעולות שציינו לעיל.	שם הפעולה תמיד באותיות קטנות, והוא אחת מ- 16 הפעולות שציינו לעיל.
לאחר שם הפעולה עם לפחות רווח/טאב אחד יופיעו האופרנדים אם יש.	לאחר שם הפעולה עם לפחות רווח/טאב אחד יופיעו האופרנדים אם יש.
רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.	רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.
תווית:	
סמל שמוגדר בתחילת משפט הוראה או הנחיית data/string. מתחילה באות אלפביתית, אח"כ אותיות אלפביות קטנות או גדולות ו/או מספרים.	סמל שמוגדר בתחילת משפט הוראה או הנחיית data/string. מתחילה באות אלפביתית, אח"כ אותיות אלפביות קטנות או גדולות ו/או מספרים.
אורך מקסימלי - 31	אורך מקסימלי - 31
סוף הגדרת שם תווית: ":" לא נכלל בתווית עצמה, חייב להיות צמוד ללא רווח.	סוף הגדרת שם תווית: ":" לא נכלל בתווית עצמה, חייב להיות צמוד ללא רווח.
אסור שתווית תוגדר יותר מפעם אחת.	אסור שתווית תוגדר יותר מפעם אחת.
אותיות קטנות וגדולות נחשבות שונות.	אותיות קטנות וגדולות נחשבות שונות.
מילים שמורות של השפה אסורות לשימוש.	מילים שמורות של השפה אסורות לשימוש.
ערך התווית - תלוי בהקשר (תווית הנחיה תקבל ערך DC נוכחי, תווית הוראה תקבל ערך IC נוכחי).	ערך התווית - תלוי בהקשר (תווית הנחיה תקבל ערך DC נוכחי, תווית הוראה תקבל ערך IC נוכחי).

סוג המשפט	הסבר כללי
משפט ריק	זוהי שורה המכילה אך ורק תווים לבנים (whitespace). כלומר מכילה רק את התווים ' ' ו- ' ' (טאבים ורווחים). ייתכן ובשורה אין אף תו (למעט התו null, כלומר השורה ריקה).
משפט הערה	זוהי שורה בה התו הראשון הוא ' '; (נקודה פסיק). על האסמבלר להתעלם לחלוטין משורה זו.
משפט הנחיה	זהו משפט המנחה את האסמבלר מה עליו לעשות כשהוא פועל על תכנית המקור. יש מספר סוגים של משפטי הנחיה. משפט הנחיה עשוי לגרום להקצאת זיכרון ואתחול משתנים של התכנית, אך הוא אינו מייצר קידוד של הוראות מכונה המיועדות לביצוע בעת ריצת התוכנית.
משפט הוראה	זהו משפט המייצר קידוד של הוראות מכונה לביצוע בעת ריצת התוכנית. המשפט מורכב משם ההוראה (פעולה) שעל המעבד לבצע, והאופרנדים של ההוראה.
משפט הנחיה:	
תווית (אופציונלי), שם ההנחיה (מתחיל ב-. וכולו אותיות קטנות), פרמטרים.	
ללא ARE, תופס את כל 14 הסיביות של המילה.	
1. data - מספרים שלמים חוקיים מופרדים ע"י פסיק. רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.	1. data - מספרים שלמים חוקיים מופרדים ע"י פסיק. רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.
ללא פסיק לפני או אחרי אוסף הפרמטרים.	ללא פסיק לפני או אחרי אוסף הפרמטרים.
משמש להקצאת תאי זיכרון לאחסון הנתונים, לאחריו מתבצע קידום מונה הנתונים בהתאם למספר הערכים.	משמש להקצאת תאי זיכרון לאחסון הנתונים, לאחריו מתבצע קידום מונה הנתונים בהתאם למספר הערכים.
אם ישנה תווית אז היא נכנסת אל טבלת הסמלים עם ערך מונה הנתונים לפני הקידום (וזוהו בעצם מיקום הנתון הראשון בסט שהתווסף).	אם ישנה תווית אז היא נכנסת אל טבלת הסמלים עם ערך מונה הנתונים לפני הקידום (וזוהו בעצם מיקום הנתון הראשון בסט שהתווסף).
2. string - פרמטר בודד - מחזרות ascii עם מרכאות. כל תו הופך למילה בפני עצמה.	2. string - פרמטר בודד - מחזרות ascii עם מרכאות. כל תו הופך למילה בפני עצמה.
בסוף מתווסף הערך 0 (\0) שמסמן את סוף המחזרות.	בסוף מתווסף הערך 0 (\0) שמסמן את סוף המחזרות.
תווית - בדומה למה שקורה בדאטה, מסמלת את כתובת תחילת המחזרות.	תווית - בדומה למה שקורה בדאטה, מסמלת את כתובת תחילת המחזרות.
3. entry - פרמטר אחד - שם תווית שמקבלת את ערכה בקובץ הנוכחי ומאפשרת לקבצים אחרים להשתמש בה כאופרנד של הוראה.	3. entry - פרמטר אחד - שם תווית שמקבלת את ערכה בקובץ הנוכחי ומאפשרת לקבצים אחרים להשתמש בה כאופרנד של הוראה.
לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.	לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.
4. extern - פרמטר אחד - שם של תווית שמוגדרת בקובץ אחר ונרצה להשתמש בה פה. הכיוון ההפוך של entry. לא ניתן להגדיר את 2 ההנחיות האלה על אותה תווית באותו קובץ.	4. extern - פרמטר אחד - שם של תווית שמוגדרת בקובץ אחר ונרצה להשתמש בה פה. הכיוון ההפוך של entry. לא ניתן להגדיר את 2 ההנחיות האלה על אותה תווית באותו קובץ.
לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.	לא ניתן להצמיד לפני כן תווית למיקום ההנחיה הזו. נוציא הודעת אזהרה - האסמבלר יתעלם מתווית זו.
משפט הוראה	
מורכב מ:	
4. תווית (אופציונלי) - אם קיימת תוכנס לטבלת הסמלים וערכה יהיה מען המילה הראשונה של ההוראה בתוך הקוד שבונה האסמבלר	4. תווית (אופציונלי) - אם קיימת תוכנס לטבלת הסמלים וערכה יהיה מען המילה הראשונה של ההוראה בתוך הקוד שבונה האסמבלר
5. שם הפעולה	5. שם הפעולה
6. אופרנדים אם אמורים להיות.	6. אופרנדים אם אמורים להיות.
שם הפעולה תמיד באותיות קטנות, והוא אחת מ- 16 הפעולות שציינו לעיל.	שם הפעולה תמיד באותיות קטנות, והוא אחת מ- 16 הפעולות שציינו לעיל.
לאחר שם הפעולה עם לפחות רווח/טאב אחד יופיעו האופרנדים אם יש.	לאחר שם הפעולה עם לפחות רווח/טאב אחד יופיעו האופרנדים אם יש.
רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.	רווחים וטאבים חופשי, פסיק אחד בודד בין כל פרמטר בלבד.
תווית:	
סמל שמוגדר בתחילת משפט הוראה או הנחיית data/string. מתחילה באות אלפביתית, אח"כ אותיות אלפביות קטנות או גדולות ו/או מספרים.	סמל שמוגדר בתחילת משפט הוראה או הנחיית data/string. מתחילה באות אלפביתית, אח"כ אותיות אלפביות קטנות או גדולות ו/או מספרים.
אורך מקסימלי - 30	אורך מקסימלי - 30
סוף הגדרת שם תווית: ":" לא נכלל בתווית עצמה, חייב להיות צמוד ללא רווח.	סוף הגדרת שם תווית: ":" לא נכלל בתווית עצמה, חייב להיות צמוד ללא רווח.
אסור שתווית תוגדר יותר מפעם אחת.	אסור שתווית תוגדר יותר מפעם אחת.
אותיות קטנות וגדולות נחשבות שונות.	אותיות קטנות וגדולות נחשבות שונות.
מילים שמורות של השפה אסורות לשימוש.	מילים שמורות של השפה אסורות לשימוש.
ערך התווית - תלוי בהקשר (תווית הנחיה תקבל ערך DC נוכחי, תווית הוראה תקבל ערך IC נוכחי).	ערך התווית - תלוי בהקשר (תווית הנחיה תקבל ערך DC נוכחי, תווית הוראה תקבל ערך IC נוכחי).

איפיון שדות
במשפטים של
שפת האסמבלר

<p>מספר: יכול להתחיל בסימן +/-, לאחר מכן סדרה של מספרים בבסיס עשרוני. אין תמיכה בבסיסים אחרים, אין תמיכה בלא שלמים. כאשר הוא מופיע כאופרנד או פרמטר של הוראה יש לו פריפיקס #.</p> <p>מחרוזת: סדרת תווי ascii מוקפת במרכאות כפולות (ואינן חלק מהמחרוזת).</p>	<p>מספר: יכול להתחיל בסימן +/-, לאחר מכן סדרה של מספרים בבסיס עשרוני. אין תמיכה בבסיסים אחרים, אין תמיכה בלא שלמים. ללא פריפיקס.</p> <p>מחרוזת: סדרת תווי ascii מוקפת במרכאות כפולות (ואינן חלק מהמחרוזת).</p>	
<p>אותו דבר למעט ההבדל בכתיבה: mcr לעומת mcro</p>	<p>לפני שהאסמבלר מתחיל לעבד את הקוד, שלב הקדם-אסמבלר מבצע פרישת מקרואים. בכל מקום בו מאוזכר שם של מאקרו שהוגדר בקובץ קודם לכן תתבצע הצבה של קטע הקוד שהמאקרו מבצע. לבסוף יוצא קובץ am. אם אין בכלל מאקרואים אז קובץ am יהיה זהה לקובץ המקור (as). אלגוריתם מוצע לקדם אסמבלר: 1. קרא את השורה הבאה מקובץ המקור. אם נגמר הקובץ עבור ל-9 (סיום) 2. האם השדה הראשון הוא שם מאקרו המופיע בטבלת המאקרו (בנוי m1)? אם כן, החלף את שם המאקרו והעתק במקומו את כל השורות המתאימות מהטבלה לקובץ, חזור ל-1. אחרת, המשיך. 3. האם השדה הראשון הוא "mcr"? אם לא, עבור ל-6. 4. הדלק דגל "יש מאקרו" 5. הכנס לטבלת שורות מאקרו את שם המאקרו. 6. קרא את השורה הבאה מקובץ המקור. אם נגמר קובץ המקור, עבור ל-9. אם דגל "יש מאקרו" דולק ולא זוהתה תויות endmcr הכנס את השורה לטבלת המאקרו ומחק את השורה הנ"ל מהקובץ. אחרת, (לא מאקרו) חזור ל-1. 7. האם זוהתה תויות endmcr? אם כן, מחק את התויות מהקובץ והמשיך. אם לא, חזור ל-6. 8. כבה דגל "יש mcr". חזור ל-1. 9. סיום: שמירת קובץ מקור פרוש.</p>	<p>קדם אסמבלר</p>
<p>מספר ראשון: זיהוי סמלים (תויות), בניית טבלת ערכים ונתינת ערך מספרי דצימלי לסמל (המען בזיכרון שהסמל מייצג), לאחר מכן הצבת הערך המספרי בקוד בכל מקום בו הסמל מופיע כאופרנד. נבצע ספירה של המקומות בזיכרון שההוראות תופסות (כל הוראה נטענת למקום העוקב להוראה הקודמת). הספירה תציין את מען ההוראה הבאה. הספירה תיעשה בשלב זה ותאוחסן במונה ההוראות IC. ערך התחלתי: 0, מיקום בפועל בקוד: 0 + 100 (100 הוגדר להיות המקום ההתחלתי בזיכרון לכל התוכנית). ה-IC מתעדכן בכל שורת הוראה המקצה מקום בזיכרון. לאחר שהאסמבלר קובע מהו אורך ההוראה, ה-IC מוגדל במספר התאים (מילים) הנתפסים על ידי ההוראה, וכך הוא מצביע על התא הפנוי הבא. טיפ מוסווה: שימוש enumsb לאחסון ושליפה מהירה של קידודי שמות פעולה. בחוברת כתוב "טבלה" אבל ברור שזו הכוונה. לכל פקודה - נסרוק את הפקודה בשלמותה ונקודד את המילה הראשונה וכן את המילים הבאות (אם יש) לפי האופרנדים. כך גם נדע מה גודל ההוראה שייטען ל-IC. כשהאסמבלר נתקל בתויות בתחילת שורה הוא רושם את התויות בטבלת הסמלים ומשייך לה את תוכנו הנוכחי של ה-IC לפני קידום. למעשה בשלב זה בונים קידוד בינארי מלא של המילה הראשונה של כל הוראה, של מילת מידע נוספת של אופרנד מיידי או רגיסטר, וכן את הקידוד הבינארי של כל הנתונים.</p> <p>מספר שני: בניית קוד המכונה באמצעות ערכי הסמלים, קודי הפעולה ומספרי האוגרים. בתכלס כיוון שהתאפשר במעבר הראשון לדחוס גם את המרת מספרי האוגרים וקודי הפעולה אז כאן זה נטו השלמה של האופרנדים החסרים שהיו תויות. בסוף השלב הזה אפשר לבצע קידוד לבסיס הנדרש ואז התוכנית מתורגמת בשלמותה כבר לקוד מכונה.</p>	<p>מספר ראשון: זיהוי סמלים (תויות), בניית טבלת ערכים ונתינת ערך מספרי דצימלי לסמל (המען בזיכרון שהסמל מייצג), לאחר מכן הצבת הערך המספרי בקוד בכל מקום בו הסמל מופיע כאופרנד. נבצע ספירה של המקומות בזיכרון שההוראות תופסות (כל הוראה נטענת למקום העוקב להוראה הקודמת). הספירה תציין את מען ההוראה הבאה. הספירה תיעשה בשלב זה ותאוחסן במונה ההוראות IC. ערך התחלתי: 0, מיקום בפועל בקוד: 0 + 100 (100 הוגדר להיות המקום ההתחלתי בזיכרון לכל התוכנית). ה-IC מתעדכן בכל שורת הוראה המקצה מקום בזיכרון. לאחר שהאסמבלר קובע מהו אורך ההוראה, ה-IC מוגדל במספר התאים (מילים) הנתפסים על ידי ההוראה, וכך הוא מצביע על התא הפנוי הבא. טיפ מוסווה: שימוש enumsb לאחסון ושליפה מהירה של קידודי שמות פעולה. בחוברת כתוב "טבלה" אבל ברור שזו הכוונה. לכל פקודה - נסרוק את הפקודה בשלמותה ונקודד את המילה הראשונה וכן את המילים הבאות (אם יש) לפי האופרנדים. כך גם נדע מה גודל ההוראה שייטען ל-IC. כשהאסמבלר נתקל בתויות בתחילת שורה הוא רושם את התויות בטבלת הסמלים ומשייך לה את תוכנו הנוכחי של ה-IC לפני קידום. למעשה בשלב זה בונים קידוד בינארי מלא של המילה הראשונה של כל הוראה, של מילת מידע נוספת של אופרנד מיידי או רגיסטר, וכן את הקידוד הבינארי של כל הנתונים.</p> <p>מספר שני: בניית קוד המכונה באמצעות ערכי הסמלים, קודי הפעולה ומספרי האוגרים. בתכלס כיוון שהתאפשר במעבר הראשון לדחוס גם את המרת מספרי האוגרים וקודי הפעולה אז כאן זה נטו השלמה של האופרנדים החסרים שהיו תויות. בסוף השלב הזה אפשר לבצע קידוד לבסיס הנדרש ואז התוכנית מתורגמת בשלמותה כבר לקוד מכונה.</p>	<p>אסמבלר עם 2 מעברים</p>
<p>חייבת להתבצע הפרדה בין ההוראות והנתונים בקוד המכונה (אין הכרח בקוד המקור). אין נבצע? IC סופר ומארגן את שיבוץ ההוראות, DC סופר ומארגן את שיבוץ הנתונים. כשנגיע לשלב ההמרה נשבץ תחילה את ההוראות בעזרת ה-IC + INITIAL_VAL ולאחר מכן נשבץ את הנתונים בעזרת ה-DC + INITIAL_VAL.</p>	<p>חייבת להתבצע הפרדה בין ההוראות והנתונים בקוד המכונה (אין הכרח בקוד המקור). אין נבצע? IC סופר ומארגן את שיבוץ ההוראות, DC סופר ומארגן את שיבוץ הנתונים. כשנגיע לשלב ההמרה נשבץ תחילה את ההוראות בעזרת ה-IC + INITIAL_VAL ולאחר מכן נשבץ את הנתונים בעזרת ה-DC + INITIAL_VAL.</p>	<p>הפרדת הוראות ונתונים</p>
<p>אותו דבר עד לטבלה.</p>	<p>הנחת יסוד: אין שגיאות במאקרו, לכן בקדם אסמבלר אין שלב גילוי שגיאות. לעומת זאת, באסמבלר נכסה שגיאות תחביר של תוכנית המקור (פעולה שלא קיימת, מספר אופרנדים שגוי, סוג אופרנד שלא מתאים לפעולה, שם אוגר לא תקין וכו'). כמו כן - וידיוי שכל סמל מוגדר פעם אחת בלבד. בדיקת השגיאות אמורה להתבצע בעיקר על שורה. אם יש שגיאה בקוד שמוגדר במקור השגיאה עלולה להיות מוכפלת ולא נתן בשלב זה כבר לחסוך גילויי שגיאה כפולים.</p>	<p>גילוי שגיאות בתוכנית המקור</p>

<p>שיטות מיעון חוקיות לאופרנד:</p> <table border="1"> <thead> <tr> <th>שיטות מיעון חוקיות עבור אופרנד היעד</th><th>שיטות מיעון חוקיות עבור אופרנד המקור</th><th>שם ההוראה</th></tr> </thead> <tbody> <tr><td>1,2,3</td><td>0,1,2,3</td><td>mov</td></tr> <tr><td>0,1,2,3</td><td>0,1,2,3</td><td>cmp</td></tr> <tr><td>1,2,3</td><td>0,1,2,3</td><td>add</td></tr> <tr><td>1,2,3</td><td>0,1,2,3</td><td>sub</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>not</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>clr</td></tr> <tr><td>1,2,3</td><td>1,2</td><td>lea</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>inc</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>dec</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>jmp</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>bne</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>red</td></tr> <tr><td>0,1,2,3</td><td>אין אופרנד מקור</td><td>prn</td></tr> <tr><td>1,2,3</td><td>אין אופרנד מקור</td><td>jsr</td></tr> <tr><td>אין אופרנד יעד</td><td>אין אופרנד מקור</td><td>rts</td></tr> <tr><td>אין אופרנד יעד</td><td>אין אופרנד מקור</td><td>stop</td></tr> </tbody> </table>	שיטות מיעון חוקיות עבור אופרנד היעד	שיטות מיעון חוקיות עבור אופרנד המקור	שם ההוראה	1,2,3	0,1,2,3	mov	0,1,2,3	0,1,2,3	cmp	1,2,3	0,1,2,3	add	1,2,3	0,1,2,3	sub	1,2,3	אין אופרנד מקור	not	1,2,3	אין אופרנד מקור	clr	1,2,3	1,2	lea	1,2,3	אין אופרנד מקור	inc	1,2,3	אין אופרנד מקור	dec	1,2,3	אין אופרנד מקור	jmp	1,2,3	אין אופרנד מקור	bne	1,2,3	אין אופרנד מקור	red	0,1,2,3	אין אופרנד מקור	prn	1,2,3	אין אופרנד מקור	jsr	אין אופרנד יעד	אין אופרנד מקור	rts	אין אופרנד יעד	אין אופרנד מקור	stop	<p>הודעת שגיאה תודפס לstdout (פלט סטנדרטי). יצוין מספר השורה בקובץ המקור (השורות מתחילות מ-1).</p> <p>גם אם התגלתה תקלה לא עוצרים בזיהוי התקלות, ממשיכים לרוץ על בדיקת התוכנית. אם יש שגיאות לא מייצרים את קבצי הפלט.</p> <p>שיטות מיעון חוקיות לאופרנד:</p> <table border="1"> <thead> <tr> <th>שם ההוראה</th><th>שיטות מיעון חוקיות עבור אופרנד המקור</th><th>שיטות מיעון חוקיות עבור אופרנד היעד</th></tr> </thead> <tbody> <tr><td>mov</td><td>1,3,5</td><td>3,5</td></tr> <tr><td>cmp</td><td>1,3,5</td><td>1,3,5</td></tr> <tr><td>add</td><td>1,3,5</td><td>3,5</td></tr> <tr><td>sub</td><td>1,3,5</td><td>3,5</td></tr> <tr><td>not</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>clr</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>lea</td><td>3</td><td>3,5</td></tr> <tr><td>inc</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>dec</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>jmp</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>bne</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>red</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>prn</td><td>אין אופרנד מקור</td><td>1,3,5</td></tr> <tr><td>jsr</td><td>אין אופרנד מקור</td><td>3,5</td></tr> <tr><td>rts</td><td>אין אופרנד מקור</td><td>אין אופרנד יעד</td></tr> <tr><td>stop</td><td>אין אופרנד מקור</td><td>אין אופרנד יעד</td></tr> </tbody> </table>	שם ההוראה	שיטות מיעון חוקיות עבור אופרנד המקור	שיטות מיעון חוקיות עבור אופרנד היעד	mov	1,3,5	3,5	cmp	1,3,5	1,3,5	add	1,3,5	3,5	sub	1,3,5	3,5	not	אין אופרנד מקור	3,5	clr	אין אופרנד מקור	3,5	lea	3	3,5	inc	אין אופרנד מקור	3,5	dec	אין אופרנד מקור	3,5	jmp	אין אופרנד מקור	3,5	bne	אין אופרנד מקור	3,5	red	אין אופרנד מקור	3,5	prn	אין אופרנד מקור	1,3,5	jsr	אין אופרנד מקור	3,5	rts	אין אופרנד מקור	אין אופרנד יעד	stop	אין אופרנד מקור	אין אופרנד יעד	<p>אלגוריתם שלדי</p>
שיטות מיעון חוקיות עבור אופרנד היעד	שיטות מיעון חוקיות עבור אופרנד המקור	שם ההוראה																																																																																																						
1,2,3	0,1,2,3	mov																																																																																																						
0,1,2,3	0,1,2,3	cmp																																																																																																						
1,2,3	0,1,2,3	add																																																																																																						
1,2,3	0,1,2,3	sub																																																																																																						
1,2,3	אין אופרנד מקור	not																																																																																																						
1,2,3	אין אופרנד מקור	clr																																																																																																						
1,2,3	1,2	lea																																																																																																						
1,2,3	אין אופרנד מקור	inc																																																																																																						
1,2,3	אין אופרנד מקור	dec																																																																																																						
1,2,3	אין אופרנד מקור	jmp																																																																																																						
1,2,3	אין אופרנד מקור	bne																																																																																																						
1,2,3	אין אופרנד מקור	red																																																																																																						
0,1,2,3	אין אופרנד מקור	prn																																																																																																						
1,2,3	אין אופרנד מקור	jsr																																																																																																						
אין אופרנד יעד	אין אופרנד מקור	rts																																																																																																						
אין אופרנד יעד	אין אופרנד מקור	stop																																																																																																						
שם ההוראה	שיטות מיעון חוקיות עבור אופרנד המקור	שיטות מיעון חוקיות עבור אופרנד היעד																																																																																																						
mov	1,3,5	3,5																																																																																																						
cmp	1,3,5	1,3,5																																																																																																						
add	1,3,5	3,5																																																																																																						
sub	1,3,5	3,5																																																																																																						
not	אין אופרנד מקור	3,5																																																																																																						
clr	אין אופרנד מקור	3,5																																																																																																						
lea	3	3,5																																																																																																						
inc	אין אופרנד מקור	3,5																																																																																																						
dec	אין אופרנד מקור	3,5																																																																																																						
jmp	אין אופרנד מקור	3,5																																																																																																						
bne	אין אופרנד מקור	3,5																																																																																																						
red	אין אופרנד מקור	3,5																																																																																																						
prn	אין אופרנד מקור	1,3,5																																																																																																						
jsr	אין אופרנד מקור	3,5																																																																																																						
rts	אין אופרנד מקור	אין אופרנד יעד																																																																																																						
stop	אין אופרנד מקור	אין אופרנד יעד																																																																																																						
	<p>קוד המכונה מתחלק ל"תמונת ההוראות" (הקוד) ו"תמונת הנתונים" (הדאטה). לראשון יש מונה IC, לשני יש מונה DC.</p> <p>נסמן ב-L את מספר המילים שתופס קוד מכונה של הוראה נתונה. בכל מעבר נתחיל לקרוא את הקובץ מההתחלה.</p> <p>מעבר ראשון:</p> <ol style="list-style-type: none"> אתחל $IC=0$, $DC=0$. קרא את השורה הבאה מקובץ המקור. אם נגמר קובץ המקור, עבור ל-16. האם השדה הראשון הוא סמל? אם לא, עבור ל-5. הדלק דגל "יש הגדרת סמל" האם זוהי הנחיה לאחסון נתונים, כלומר, האם הנחית data או string? אם לא, עבור ל-8. אם יש הגדרת סמל (תויות), הכנס אותו לטבלת הסמלים עם סימון (סמל מסוג data) - ערכו יהיה DC. (אם הסמל כבר נמצא בטבלה, יש להודיע על שגיאה). זוהי את סוג הנתונים, קודד אותם בזיכרון, עדכן את מונה הנתונים DC בהתאם לאורכם, חזור ל-2. האם זו הנחיית extern או הנחיית entry? אם לא, עבור ל-11. האם זוהי הנחיית extern? אם כן, הכנס כל סמל (אחד או יותר) המופיע כאופרנד של ההנחיה לתוך טבלת הסמלים ללא ערך, עם סימון (סמל מסוג external). חזור ל-2. אם יש הגדרת סמל, הכנס אותו לטבלת הסמלים עם סימון (סמל מסוג code). ערכו יהיה IC (אם הסמל כבר נמצא בטבלה יש להודיע על שגיאה). חפש את שם הפעולה בטבלת שמות הפעולות, ואם לא נמצא – הודע על שגיאה בשם ההוראה. נתח את מבנה האופרנדים של ההוראה וחשב את L. בנה כעת את הקוד הבינארי של המילה הראשונה של הפקודה. עדכן $IC + L = IC$ חזור ל-2. אם נמצאו שגיאות בקובץ המקור, עצור. עדכן בטבלת הסמלים את ערכם של הסמלים מסוג data, ע"י הוספת הערך הסופי של IC התחל מעבר שני <p>מעבר שני:</p> <ol style="list-style-type: none"> אתחל $IC = 0$ קרא את השורה הבאה מקובץ המקור. אם נגמר קובץ המקור, עבור ל-10. אם השדה הראשון הוא סמל, דלג עליו. האם זוהי הנחיית extern/string/data? אם כן חזור ל-2. האם זוהי הנחיית entry? אם לא, עבור ל-7. סמן בטבלת הסמלים את הסמלים המתאימים כ-entry. חזור ל-2. השלם את קידוד האופרנדים החל מהמילה השנייה בקוד הבינארי של ההוראה, בהתאם לשיטת המיעון. אם אופרנד הוא סמל, מצא את המען בטבלת הסמלים. עדכן $IC + L = IC$ חזור ל-2. 																																																																																																							

<p>10. אם נמצאו שגיאות במעבר שני, עצור.</p> <p>11. צור ושמו את קבצי הפלט: קובץ קוד המכונה קובץ סמלים חיצוניים, וקובץ סמלים של נקודות כניסה.</p>	
<p>בהפעלת האסמבלר מעבירים באמצעות ארגומנטים של שורת הפקודה רשימה של קבצי מקור (אחד או יותר) שהסיימת שלהם היא as, אבל בכתיבה שלהם כארגומנט המשתמש נדרש לרשום את שמם ללא הסיימת. אלו קבצי טקסט עם תוכניות בתחביר של שפת האסמבלי שלנו. האסמבלר פועל על כל קובץ בנפרד ויוצר עבורו את קבצי הפלט:</p> <ol style="list-style-type: none"> 1. קובץ am - קובץ המקור לאחר שלב הקדם-אסמבלר (פרישת המקרואים). 2. קובץ ob - קוד המכונה הסופי 3. קובץ ext - משתנים חיצוניים והכתובות בהן נעשה בהן שימוש. נוצר רק אם יש לפחות משתנה חיצוני אחד. 4. קובץ ent - מכיל את כל הסמלים שהוגדרו בקובץ זה כניתנים לשימוש בקבצים אחרים ולצידם הכתובת בה הסמל מוגדר. נוצר רק אם יש לפחות משתנה אנטרי אחד. <p>לתשומת לב: אם בקובץ המקור אין הצהרת extern. אזי לא ייווצר עבורו קובץ ext. בדומה, אם אין בקובץ המקור הצהרת entry, לא ייווצר קובץ ent. אין ליצור קובץ ext או ent שנשאיר ריק.</p> <p>הערה: אין חשיבות לסדר השורות בקבצים מסוג ent או ext. כל שורה עומדת בפני עצמה.</p>	<p>קבצי קלט ופלט של האסמבלר</p>
<p>אותו דבר למעט:</p> <p>גודלה של כניסה במערך זהה לגודלה של מילת מכונה: 14 סיביות – משום מה כתוב 10 סיביות בחוברת, מנחש שזו טעות.</p> <p>האסמבלר מחזיק שני מערכים - מערך ההוראות ומערך הנתונים. מערכים אלו נותנים למעשה תמונה של זיכרון המכונה (גודל כל כניסה במערך זהה לגודלה של מילת מכונה: 12 סיביות). במערך ההוראות מכנים האסמבלר את הקידוד של הוראות המכונה. במערך הנתונים מכנים האסמבלר את קידוד הנתונים (שורות מסוג .data. string).</p> <p>לאסמבלר יש שני מונים: מונה ההוראות (IC) ומונה הנתונים (DC). מונים אלו מצביעים על המקום הבא הפנוי במערכים לעיל. כשמתחיל האסמבלר לעבור על קובץ מקור, שני מונים אלו מאופסים.</p> <p>בנוסף יש לאסמבלר טבלה, לכל התוויות בהן נתקל האסמבלר במהלך המעבר על הקובץ - טבלת סמלים (table-symbol). לכל סמל (תווית) נשמרים שמו, ערכו וטיפוסו (external או relocatable).</p> <p>האסמבלר קורא את קובץ המקור שורה אחר שורה, מחליט מהו סוג השורה (הערה, הוראה, הנחיה או שורה ריקה) ופועל בהתאם.</p> <ol style="list-style-type: none"> 1. שורה ריקה/הערה – מתעלם וממשיך הלאה 2. שורת הוראה - מזהה את הפעולה ואת שיטות המיעון של האופרנדים, וקובע את ערך האופרנדים כך: <ul style="list-style-type: none"> • אם האופרנד הוא רגיסטר – מספר הרגיסטר • תווית (מיעון ישיר) – ערך התווית כמו שמופיע בטבלת הסמלים • מספר (מיעון מיידי) - המספר עצמו האופרנד מזוהה לפי התחביר שלו. <p>בשלב הבא האסמבלר מכניס למערך ההוראות במקום עליו מצביע IC את קוד המילה הראשונה (לאחר קידוד). מילה זו מכילה את קוד הפעולה, ואת שיטות המיעון.</p> <p>לאחר מכן משריין האסמבלר מקום במערך עבור המילים הנוספות הנדרשות עבור ההוראה, אם נדרשות כאלה, ומגדיל את מונה ההוראות בהתאם.</p> <p>אם אלו מילים שניתן לקודד מיד (כלומר לא תווית) אז הוא כבר מקודד ומכניס אותם למערך.</p> <p>אם בשורת ההוראה קיימת תווית, אזי התווית מוכנסת אל טבלת הסמלים תחת השם המתאים, ערך התווית הוא ערך מונה ההוראות לפני קידוד ההוראה, וסוג התווית הוא relocatable.</p> <ol style="list-style-type: none"> 3. שורת הנחיה – <ul style="list-style-type: none"> • Data - האסמבלר קורא את רשימת המספרים, מכניס אותם אחד אחד אל מערך הנתונים, ומקדם את DC באחד עבור כל מספר שהוכנס. אם יש תווית היא מוכנסת לטבלת הסמלים ומקבלת את ערכו של DC לפני קידום. הטיפוס של התווית הוא R ומסומן שהוא הוגדר 	<p>אופן פעולת האסמבלר</p>

	<p>בחלק הנתונים.</p> <p>בסוף המעבר הראשון ערך התווית יעודכן בטבלת הסמלים ע"י הוספת ערך IC לערכו (ואז מתקבל המיקום הסופי האמיתי של התווית).</p> <ul style="list-style-type: none"> String – כמו הדאטה, רק שכאן קודי ascii של המחזות מוכנסים אל מערך הנתונים כל אחד בנפרד + הערך "0" בסוף. מונה הנתונים מקודם במספר הערכים שהוכנסו, כלומר אורך המחזות + 1 (עבור 0). תווית – בדומה לדאטה. Entry – האסמבלר רושם את הבקשה להכניס את התווית המוכרזת אל קובץ entn. האסמבלר רושם את הבקשה ובסיום העבודה התווית תירשם בקובץ entn. Extern – הצהרה על סמל חיצוני שהקוד בקובץ המקור עושה בו שימוש. האסמבלר מכניס את הסמל לטבלת הסמלים. ערכו הוא 0 וטיפוסו E. לא ידוע איפה הוא מוגדר במקור וזה גם לא משנה לאסמבלר. <p>אפשר להשתמש בשם של סמל גם אם ההצהרה עליו מגיעה רק בהמשך הקובץ (גם אם מדובר בסמל חיצוני).</p>	
<p>אותו דבר למעט בסיס ההצגה:</p> <p>ההצגה של שורות הקוד תתבצע בקידוד הבינארי הייחודי "/" ו"0" שהוגדר לעיל.</p>	<p>קידוד ההוראה הראשונה יכנס למען 100 (בבסיס 10) בזיכרון, קידוד ההוראה השנייה יכנס למען העוקב (תלוי במספר המילים של ההוראה הראשונה), וכך הלאה עד להוראה האחרונה.</p> <p>מיד לאחר קידוד ההוראה האחרונה, מכניסים לתמונת הזיכרון את קידוד הנתונים ('data' string). הנתונים יוכנסו בסדר בו הם מופיעים בקובץ המקור. אופרנד של הוראה שמתייחס לסמל שהוגדר באותו קובץ, יקוד כך שיצביע על המקום המתאים בתמונת הזיכרון שבונה האסמבלר.</p> <p>נשים לב שהמשתנים מופיעים בתמונת הזיכרון <u>אחרי ההוראות</u>. זוהי הסיבה בגללה יש לעדכן בטבלת הסמלים, בסוף המעבר הראשון, את ערכי הסמלים המגדירים נתונים (סמלים מסוג data).</p> <p>עקרונית, קובץ object מכיל את תמונת הזיכרון שתוארה כאן. קובץ object מורכב משורות של טקסט כדלקמן:</p> <p>השורה הראשונה היא כותרת המכילה שני מספרים בבסיס 10: האורך הכולל של קטע ההוראות (במילות זיכרון) ואחריו האורך הכולל של קטע הנתונים (במילות זיכרון). בין שני המספרים יש רווח אחד.</p> <p>השורות הבאות מכילות את תוכן הזיכרון. בכל שורה מופע תוכן של מילה אחת, לפי הסדר החל מהמילה בכתובת 100. תוכן המילה מקודד בשיטת Base64.</p> <p>https://en.wikipedia.org/wiki/Base64</p> <p>מילה היא בגודל 12 ביטים. כל 6 ביטים יומרו לתו מתאים לפי הטבלה המגדירה את base 64.</p> <p>לכן בכל השורות מלבד השורה הראשונה יהיו בדיוק 2 תווים בקוד base64.</p>	<p>פורמט קובץ ה-object</p>
<p>אותו דבר למעט בסיס ההצגה:</p> <p>בהנחיות בחוברת יש הנחיה סותרת. במקום אחד כתוב שצריך להיות מיוצג בבסיס 2 הייחודי (/ או .) ובמקום אחר מוצגת דוגמה לקובץ סופי בייצוג עשרוני. יש בין הפרוייקטים של הסמסטר הקודם כאלה שעשו ייצוג עשרוני אז מי שנעזר בפרוייקטים אלו כדי לקבל תמונה של איך אמור להיראות פרוייקט שישים לב כי זה מבלבל.</p>	<p>כל שורה מכילה שם של סמל שהוגדר כ- entry ואת ערכו כפי שנמצא בטבלת הסמלים. הערכים מיוצגים בבסיס 10.</p>	<p>פורמט קובץ ה-entries</p>
<p>אותו דבר למעט בסיס ההצגה:</p> <p>בהנחיות בחוברת יש הנחיה סותרת. במקום אחד כתוב שצריך להיות מיוצג בבסיס 2 הייחודי (/ או .) ובמקום אחר מוצגת דוגמה לקובץ סופי בייצוג עשרוני. יש בין הפרוייקטים של הסמסטר הקודם כאלה שעשו ייצוג עשרוני אז מי שנעזר בפרוייקטים אלו כדי לקבל תמונה של איך אמור להיראות פרוייקט שישים לב כי זה מבלבל.</p>	<p>כל שורה מכילה שם של סמל שהוגדר external, וכתובת בקוד המכונה בה יש קידוד של אופרנד המתייחס לסמל זה. כמובן שיתכן ויש מספר כתובות בקוד המכונה בהם מתייחסים לאותו סמל חיצוני. לכל התייחסות כזו תהיה שורה נפרדת בקובץ ה-externals. הכתובות מיוצגות בבסיס 10.</p> <p>ייתכן ויש מספר כתובות בקוד המכונה בהן מילות המידע מתייחסות לאותו סמל חיצוני. לכל כתובת כזו תהיה שורה נפרדת בקובץ ה-externals.</p>	<p>פורמט קובץ ה-externals</p>

<p>סיכום והנחיות כלליות</p>	<ul style="list-style-type: none">• אורך התוכנית, הניתנת כקלט לאסמבלר אינו ידוע מראש, ולכן אורך התוכנית המתורגמת אינו אמור להיות צפוי מראש. אולם כדי להקל בשימוש האסמבלר, ניתן להניח גודל מקסימלי. למיכך יש אפשרות לחששות במערכים לאבסון תמונת קוד המכונה בלבד כל מבנה נתונים אחר (למשל טבלת הסמלים וטבלת המקור), יש לממש באופן יעיל והסכני (למשל באמצעות רשימה מקושרת והקצאת זיכרון דינאמי).• השמות של קבצי הפלט צריכים להיות תואמים לשם קובץ הקלט. למעט הסיומות. למשל, אם קובץ הקלט הוא prog.as אזי קבצי הפלט שיווצרו הם : prog.ob, prog.ext, prog.ent• מתכנת הפעלת האסמבלר צריכה להיות כפי הנדרש במיין, ללא שינויים כלשהם. כלומר, ממשק המשתמש יהיה אך ורק באמצעות שורת הפקודה. בפרט, שמות קבצי המקור יועברו לתוכנית האסמבלר כארגומנטים (לאחד או יותר) בשורת הפקודה. אין להוסיף תפריטי קלט אינטראקטיביים, חלוטות גרפיים למיניהם, וכד'.• יש להקפיד לחלק את מימוש האסמבלר למספר מודולים (קבצים בשפת C) לפי משימות. אין לרכז משימות מסוגים שונים במודול יחיד. מומלץ לחלק למודולים כגון : מעבר ראשון, מעבר שני, פונקציות עזר (למשל, תרגום לבסיס, ניתוח תחבירי של שורה), טבלת הסמלים, מפת הזיכרון, טבלאות קבועות (קודי הפעלה, שיטות המיפון החוקיות לכל פעולה, וכד').• יש להקפיד ולתעד את המימוש באופן מלא וברור, באמצעות הערות מפורטות בקוד.• יש לאפשר תווים לבנים עודפים בקובץ הקלט בשפת אסמבלר. למשל, אם בשורת הוראה יש שני אופרנדים המופרדים בפיסק, אזי לפני ואחרי הפסיק מותר שהיו רווחים וטאבים בכל כמות. בדומה, גם לפני ואחרי שם הפעלה. מותרות גם שורות ריקות. האסמבלר יתעלם מתווים לבנים מיותרים (כלומר ידלג עליהם).• הקלט (קוד האסמבלר) עשוי להכיל שגיאות תחביריות. על האסמבלר לגלות ולדווח על כל השגרות השגויות בקלט. אין לעצור את הטיפול בקובץ קלט לאחר גילוי השגיאה הראשונה. יש להדפיס למסך הודעות מפורטות ככל הניתן, כדי שאפשר יהיה לחבין מה וחיכן כל שגיאה. כמובן שאם קובץ קלט מכיל שגיאות, אין טעם להפיק עבורו את קבצי הפלט (ob, ext, ent). <p>תם ונשלם פרק החסברים והגדרת הפרויקט.</p> <p>בשאלות ניתן למנות לקבוצת הדיון באתר הקורס, ואל כל אחד מהמנחים בשעות הקבלה שלהם.</p> <p>להזכירכם, באפשרותו של כל סטודנט למנות לכל מנחה, לאו דווקא למנחה הקבוצה שלו, לקבלת עזרה. שוב מומלץ לכל אלה שטרם בדקו את התכנים באתר הקורס לעשות זאת. נשאלות באתר זה הרבה שאלות בנושא חומר הלימוד והממיינים, והתשובות יכולות להועיל לכולם.</p> <p>לתשומת לבכם : לא תיתן דחייה בהגשת המיין, פרט למקרים מיוחדים כגון מילואים או אשפוז. במקרים אלו יש לבקש ולקבל אישור מראש ממנחה הקבוצה.</p>
<p>חידודים מפורום הקורס (2023)</p> <p>באדום – הערות של אסתר שמאשרות סופית מה נדרש בשאלות בהן לא הייתה הכרעה ברורה.</p>	<p>תוויות</p> <ol style="list-style-type: none">1. תווית יכולה להכיל בתוכה שם של פקודה כל עוד היא תת מחרוזת (לדוגמה h2mov).2. יכולות להיות data או string. ללא תווית. גם שם כמובן מעלים את המונה.3. תווית לא יכולה להיות מוגדרת גם בקובץ וגם בחיצוני.4. הגדרת תווית יכולה להתבצע פעם אחת בלבד (שם תווית ולאחר מכן נקודותיים).5. תווית יכולה להיות מוגדרת ללא אפיון של entry או extern. תווית שמוגדרת כentry היא תווית ככל התווית אלא שניתן להשתמש בה בקבצים אחרים.6. תווית שהוגדרה כ entry אך לא הוכרזה באופן עצמאי -> שגיאה, כמו תווית שנקראו ולא הוכרזו קודם. entry לא מכריז על תווית אלא רק מציין שאפשר להשתמש בה אח"כ בקבצים אחרים.7. אם קיימת הנחיית extern על תווית וגם הגדרה של אותה תווית באותו קובץ זו שגיאה.8. מקרה של תווית ללא הנחיה או הוראה אחריה יש להודיע על שגיאה.9. בתחילת ההוראה הראשונה בקובץ אין שום חיוב שתהיה תווית.10. אם מתגלית שגיאה בשורה שיש בה תווית בתחילתה או שמשתמשת בתווית מוציאים הודעת שגיאה באשר לשגיאה הרלוונטית וממילא אין ליצור קבצי פלט.11. תווית בשם של מאקרו – תהיה שגיאה ממילא בעיבוד ולכן אפשר ליצין בהערה בתייעוד שממילא תתקבל שגיאה. בכל זאת אפשר לנסות להכניס את זה בהודעה למשתמש אם רוצים.12. אם יש תווית שהוגדרה ואין בה שימוש – לא שגיאה. כנ"ל מקר.13. אם הייתה הגדרה של extern או entry שלא נעשה בהם שימוש - אחד המנחים ענה שזו שגיאה, אבל אסתר ענתה שזו לא שגיאה... יש הבדל בין תווית חיצונית לתווית פנימית: <p>תווית חיצונית: ניתן להצהיר (extern LABEL). על תווית כתווית חיצונית ולא להשתמש בה בכלל בפקודות ההוראה. במקרה כזה אין להוסיף את התווית לקובץ . ext . אם אין תווית חיצונית מנספות אין ליצור קובץ . ext . תווית המוצהרת כתווית חיצונית ובנוסף גם מוגדרת כתווית בקובץ זו שגיאה (extern LABEL and LABEL:).</p> <p>תווית פנימית: חובה להגדיר תווית המוצהרת כתווית פנימית, אחרת - שגיאה. לכן, אם יש שורה כזו " entry LABEL ואין שורה כזו" A " BEL:אזי יש לדווח על שגיאה. שימו לב שתוכן קבצי ext/ent מעט שונים.</p> <ol style="list-style-type: none">14. הכרזה על תווית כ- entry או extern. פעמיים – יש סתירה, כמה פעמים נכתב שיש להוציא שגיאה, כמה

פעמים נכתב שזה בסדר. **אסתר ענתה כמה פעמים:**
הנחיית entry/extern יכולה להתבצע כמה פעמים וזו
לא שגיאה (extern LABEL, extern LABEL). זו
המסקנה הסופית? אפשר להצהיר כמה פעמים על תווית
entry/extern. אפשר להוציא אזהרה.

משפטי הנחיה

15. הופעה של תווית במקום ערך מספרי ב data. למשל:
data VAL - **החוברת** הגדירה שלא **ואסתר** גם אישרה
שככה כתוב בחוברת אבל **רועי** קודם לכן אמר
שבאסמבלר תקני יש שימוש במהלך כזה ולכן נראה לו
שיש לאפשר זאת. **מאז ענו כמה פעמים שלא.** אין
לאפשר תווית בהנחיית data.
16. string ניתן להצהיר רק על מחרוזת אחת בכל פעם.
17. אי אפשר להגדיר את אותה תווית גם entry וגם
extern.
18. אחרי ההנחיות extern | entry חייב להתקבל שם של
תווית. באשר לשאלה "האם ניתן לציין כמה תוויות יחד
בהנחיות אלו" **נראה שיש סתירה.**
בעמוד 29 בחוברת מצויין שextern מקבל כפרמטר רק
תווית אחת, אבל בעמוד 36 באלגוריתם של המעבר
הראשון נראה שיש תמיכה בהצהרה על מספר תוויות
במכה אבל זה מפורש רק בextern.
רועי הבהיר בהמשך ש 36 הוא הכלל לפיו צריך לנהוג.
עמוד 29 היא הדגמה מקומית ולא ההוראה לביצוע,
כלומר יכול להיות יותר מתווית אחת.
עמוד 36 נותן תשובה רק באשר לextern, מה לגבי
entry?
אסתר ענתה בכמה פעמים שלגבי entry יש הגבלה
לתווית אחת, אך לextern אין הגבלה, וציינה שהתוויות
יהיו מופרדות באמצעות פסיקים.
חיים ענה בתגובה יותר עדכנית שמותר רק תווית אחת
בלבד ב-2 סוגי ההנחיות הנ"ל.
אסתר ענתה לאחר מכן שיש להתייחס למופיע
באלגוריתם השלדי, כלומר כאמור לextern מותר יותר
מתווית אחת.
העדכון האחרון (מה06/08) של אסתר הוא הבהרה
שדווקא כן ניתן לכתוב מספר תוויות הן לאחר פקודת
extern והן לאחר פקודת entry. ההפרדה בין התוויות
תהיה על ידי פסיק.
מתבקש שתצא הודעה נעוצה בנושא זה שמציינת
שכלל המנחים מעודכנים בנוגע להנחיה זו.
מסקנה סופית – אפשר להצהיר על מספר תוויות לאחר
הנחיית entry/extern
19. במשפט הנחיה - האם חייב להיות רווח/טאב בין תווית
(אם יש) לבין הנקודה, בין הנקודה לשם ההנחיה, בין שם
ההנחיה לפרמטרים ובמשפט הוראה האם חייב להיות
רווח בין התווית (אם יש) לשם הפעולה? **רועי ענה שלא**
חייב, אבל אני חושב שזה לא תקין. בוודאי בין הנקודה
לשם ההנחיה ובין ההנחיה לפרמטרים. האם אפשר
לחייב בכל זאת? חייב להיות תו לבן בין הנקודותיים של
התווית לנקודה של ההנחיה. לא חובה שיהיה רווח בין
התווית לשם הפעולה. יש לשים לב לתווים לבנים
אופציונאליים ולפסיקים תקינים.
20. מחרוזת חוקית היא סדרת תווי ascii נראים (שניתנים
להדפסה), המוקפים במרכאות כפולות. זה כולל תווים
לבנים (רווח, טאב). תווים שאינם נראים הם תווים
מיוחדים אחרים, אפשר לחפש בגוגל... חיים מאשר
שתווים מותרים הם החל מהערך 32 בטבלה - רווח (כולל)
עד הערך 126 - ~.
21. במחרוזת ריקה יש להכניס תו סיום מחרוזת בלבד. מה
שקיים בתוך הגרשים זה מה שיש להכניס למחרוזת.
אם יש תו / יש להכניס אותו. אין צורך להמיר ל escape
characters. אם יש מרכאות באמצע הטקסט אבל יש גם
מרכאות פותחות וסוגרות באופן תקין אפשר להחליט
לכלול את המרכאות הפנימיות כחלק מהמחרוזת ואפשר
לחילופין שלא אבל אז צריך לנהל את השגיאה.
22. במקרה של escape char במחרוזת יש להכניס את התו
באותה צורה בה הוא הוזן (לדוגמה t / יונכס כ"t").

	<p>23. אורך שורה מקסימלי 80 (לא כולל התו של השורה החדשה), ז"א שלמחרוזת צריך להקצות 81 מקומות (בפועל הכוונה מן הסתם לעשות את זה כולל מילת ההנחיה של המחרוזת).</p> <p>הוראות</p> <p>24. מה קורה אם יש לי הוראה עם שני אופרנדים, כאשר אחד מהם תווית extern ואחד תווית רגילה. האם היא תקבל R או E?</p> <p>זו שגיאה מלכתחילה. אין פקודה שמקבלת 2 אופרנדים שהם תווית...</p> <p>שורת הערה</p> <p>25. שורת הערה יכולה לבוא כשורה נפרדת בלבד. התו הראשון חייב להיות ;</p> <p>26. שורה שיש בה רווחים וטאבים לפני ; היא עדיין שורת הערה (לדלג על התווים הלבנים) – יש סתירה! רוני בן ישי ענה בפורום שלא וגם אסתר ענתה שלא, אבל מנחה אחר ענה קודם שכן...</p> <p>לעניות דעתי נראה שיש להתיר, כיוון שבשפות המצויות בשימוש נפוץ הדבר הזה מותר.</p> <p>לפי המופיע בחוברת המטלות - התו הראשון צריך להיות ";" על מנת להצהיר על שורה כשורת הערה</p> <p>מעברים וניהול נתונים</p> <p>27. עקרונית אפשר במעבר הראשון רק לשמור את הכתובות של label בטבלת הסמלים מבלי לקודד את כל שאר הדברים שניתן לקודד ובמעבר השני לקודד פשוט הכל ביחד. מה שכתוב בחוברת לגבי זה זו המלצה.</p> <p>28. אפשר להכניס במערך הנתונים ובמערך הפקודות ערכים בבסיס עשרוני (ורק כשנכתוב לקובץ נמיר לבינארי).</p> <p>29. ניתן להחזיק את הנתונים בבסיס 64 כבר כשהם מקודדים ולא לשמור אותם בצורה בינארית, בתכלס זה לא ממש קריטי.</p> <p>30. אפשר למחוק הערות ושורות ריקות כבר בשלב הקדם אסמבלר, אין שחובה לשמור אותם גם בקובץ המקור המורחב, אבל זה לבחירתנו בגדול.</p> <p>31. טבלת אחסון מקרואים – לבנות מבנה זיכרון דינאמי</p> <p>32. IC+DC צריך להיות קטן שווה ל924 (100 תאים ראשונים בזיכרון "שמורים").</p> <p>33. הזיכרון מנוהל עבור כל קובץ בנפרד.</p> <p>לכל קובץ מקור יש לאתחל את כל המשתנים.</p> <p>לכל קובץ יש מערך של נתונים ומערך של פקודות משלו.</p> <p>IC, DC מתאפסים.</p> <p>לכל קובץ מקור יש טבלת סמלים משלו.</p> <p>34. אין להכניס לטבלת הסמלים את הערך עצמו שהסמל מגדיר (רשימת המספרים או המחרוזת), הערכים נכנסים לטבלת הנתונים (data image).</p> <p>35. מימוש הייצוג הבינארי כרצוננו, אפשר כמצביע למערך של char ואפשר struct וכו', כמובן עדיף לשמור על יעילות מירבית וצריך לשמור על גבולות.</p> <p>36. בקידוד פקודה סדר הקידוד (המילים הנוספות) צריך להיות לפי סדר האופרנדים.</p> <p>37. לפי האלגוריתם המוצג בחוברת יש טבלת סמלים אחת בלבד ולכל שורה של הטבלה מכניסים סימון דאטה או קוד בהתאם לסמל (כך לפי התשובה של אסתר), אבל אם אני זוכר נכון כתוב שהאלגוריתם לא מחייב...</p> <p>38. התוכנית מקמפלט עבור 2' compliment machine. גם הקמפול עצמו יתבצע במכונה כזו (לפי אסתר). זה משנה כמובן עבור bitwise operations.</p> <p>39. אין מניעה להשתמש במערכים כל עוד אין סתירה בין השימוש ובין הצורך על פי אפיון המטלה</p> <p>40. הטווח של מספר כאופרנד בהוראה הוא 10 ביטים ולכן לו טווח אחד, ואילו עבור מספר בהנחיה הוא 12 ביטים ולכן לו טווח אחר. לשים לב למגבלת הביטים השונה!</p> <p>41. ניתן כבר בשלב הקדם אסמבלר למחוק רווחים עודפים.</p> <p>42. אין צורך להדפיס את טבלת המאקרו או את הטבלה של הסמלים.</p> <p>43. אם מתקבל קובץ ריק אין צורך ליצור קבצי פלט</p>
--	--

44. הדרישה של המטלה היא שלכל תא בזיכרון יהיה 12 BITS בדיוק, לא פחות ולא יותר.
45. אם יש שורה שארוכה מ-80 תווים אפשר להתעלם מהשורה או להתייחס ל-80 התווים הראשונים. כרצונך (אסתר). **חשוב לזכור שבמקרה כזה אין פלטים מאחר ויש שגיאה בקובץ (שורה ארוכה מ-80 תווים).**
46. **האם ספירת התווים המקסימליים בשורה לוקחת בחשבון תווים לבנים? הרי נאמר שצריך להתעלם מהם בהקשרים אחרים.**
- 80 כולל תווים לבנים**
47. אפשר להגדיר מספר חד ספרתי של משתנים גלובלים. מומלץ להימנע.

ניהול שגיאות ואזהרות

48. מספיק לגלות שגיאה אחת בשורה ולהמשיך לבאות, אין צורך להמשיך לגלות שגיאות בשורה הזו.
49. עבור שגיאה במעבר ראשון/שני קובץ amc כבר קיים ולכן נשאר אותו כמו שהוא ואילו את הקבצים האחרים לא ניצור.
50. האם צריך לבדוק שאורך שורה לא גדול מ-80 בשלב הקדם אסמבלר או במעבר הראשון?
- רועי:**
- אם זה לא כתוב בצורה חחע במטלה אז אפשר כך או אחרת רק תעד את זה בקוד. לי נראה טבעי יותר לבדוק את זה לאחר פרישת המקור.
- אסתר:**
- נראה שצריך לבדוק זאת בשלב הקדם אסמבלר (מה שאסתר אומרת נשמע הגיוני, את תוכן המקור ממילא בודקים ב-2 הוריאציות, וכל שנשאר הוא לראות אם שורת ההכרזה על המקור לא חורגת באורכה).
51. הודעת שגיאה צריכה לציין את מספר שורת המקור שבה ישנה השגיאה. **לפי אסתר – הכוונה היא לקובץ הפרוש.**
52. אם מספר המתקבל כאופרנד או כנתון דאטה חורג ממגבלת הסיביות – להוציא שגיאה.
53. אפשר בנוסף להדפסות Error להוסיף גם הדפסות Info, לדוגמה - קובץ קיים, קובץ נוצר וכו'...
54. אם יש שגיאה בשם המקור, אין ליצור קבצי פלט. אך יש לגלות שגיאות נוספות (אם יש).
55. במקרה של שגיאה בהקצאת זיכרון יש להדפיס שגיאה ולצאת לגמרי מהתוכנית.
56. במקרים של אזהרות ללא שגיאות יש להוציא קבצי פלט.
57. הנחיית data או string ללא נתונים – **סתירה: במקום אחד נאמר שיש להוציא אזהרה.**
- במקום אחר נאמר:** אם נמצאו משפטי הנחיה בלי ערך יש להודיע על שגיאה. כמו כן נאמר שאחרי ההנחיות extern ו entry חייב להתקבל שם של תווית. **זו שגיאה**
58. לאן להדפיס הודעת שגיאה? **בחוברת כתוב ל- stdout.** כך גם בפורום מספר פעמים. **אבל כתוב גם בפורום מספר פעמים** שצריך להוציא ל- stderr. **אסתר** ענתה stderr אבל גם אם נוציא לstdout לא ירד על זה ניקוד, ולאחר מכן **מנחה אחר** ענה בשרשור אחר stdout. **יותר נכון להוציא שגיאות פלט לקובץ stderr. אך נתון לבחירתכם**
59. צריך לבדוק שגיאות לוגיות (לפי אסתר – וגם לפי שאר המנחים...), לדוגמה: אם יש קוד כלשהוא אחרי הקריאה לפקודה stop זו שגיאה לוגית.
60. יש לגלות את כל השגיאות של שלב קדם אסמבלר אך אין לעבור לשלבים הבאים.
61. במצב בו מנסים ליצור קבצי ext-i ob, ent, והבקשה נכשלת אין לנסות להמשיך להפיק את הקבצים שכן מתאפשר, יש להמשיך לעיבוד של קובץ המקור הבא. מן הסתם להוציא כאן הודעת שגיאה...
62. אי אפשר להגדיר 2 מאקרואים באותו שם. זו שגיאה. אם יש דריסה של מאקרו, כלומר יש הגדרה חדשה של מאקרו שכבר הוגדר אך עם תוכן שונה יש להודיע על שגיאה, להשלים את מעבר הקדם אסמבלר על הקובץ לזיהוי שגיאות נוספות ולצאת.
63. אם קראתי לקובץ שלא קיים יש להודיע על כך ולהמשיך לקובץ הבא.

	<p>64. גם אם גילינו שגיאה במעבר הראשון יש להמשיך ולגלות את השגיאות של המעבר השני.</p> <p>65. עבור חריגת זיכרון מוציאים שגיאה ומתייחסים כמו כל שגיאה אחרת.</p> <p>מאקרו</p> <p>66. המילים השמורות הן case sensitive ככה שאפשר להשתמש במילים האלה למקראים או תוויות בעיוות.</p> <p>67. שם של מאקרו צריך לעמוד בתנאים של שם תווית חוקית</p> <p>68. קריאה למאקרו לפני הצהרתו – אפשר לעצור ולעבור לקובץ הבא (לפי רועי רחמני). לפי מיכל ניתן להניח שזה לא קורה. אפשר לציין בהערה את מה שמיכל ציינה.</p> <p>69. האם קריאה למאקרו יכולה להופיע בשורה של LABEL? m1: L1? אסתר ענתה שכן, צריך לפרוש מאקרו בכל מקום בו הוא נמצא. במקומות אחרים ענתה שאפשר להניח ששם המאקרו צריך להופיע כמילה ראשונה בשורה (תווים לבנים אפשריים לפני).</p> <p>האם הכוונה שהנושא לבחירתנו או שישנה הכרעה כלשהי בנושא? לבחירתכם. יותר קל להניח ששם מאקרו יחיד בשורה. פרישה בכל מקום תגרוור שאלות נוספות (מופיעות בפורום) וכנראה גם שגיאות במעבר הראשון.</p> <p>70. לא ניתן להגדיר מאקרו בשורה עם תווים אחרים, פרט לתווים לבנים.</p> <p>71. אין חיוב להזחה בשורות שבתוך המאקרו. זה שקוף.</p> <p>72. אם העברנו קובץ באסמבלר בהצלחה ואז שינינו אותו ונוצרה שגיאה – האם בפעם הבאה שהקובץ נכנס לאסמבלר יש צורך למחוק את הקבצים מההרצה הקודמת? לכאורה הם כבר לא בתוקף.</p> <p>אסתר: השאלה לא ברורה. כל הרצה בפני עצמה - אם התרגום עובר בהצלחה נוצרים הקבצים. אפשר לשנות את הקובץ (אם כי לא מומלץ) אך השינוי לא אמור להוביל לשינוי של הרצות קודמות. התשובה היא שלא צריך להתייחס לקבצים קודמים אם נוצרו בהרצות קודמות. יש להקפיד על פלט השגיאות..</p> <p>73. אם הוגדר מקרו עם שם ולאחריו כמובן הופיע endmacro אבל ביניהם לא היה שום טקסט, כלומר, המקרו ריק, אז אפשר פשוט להמשיך כרגיל.</p> <p>74. אם יש מקרו שהוגדר ואין בו שימוש – לא שגיאה. כנ"ל תווית.</p> <p>75. ניתן להניח שגם אין שימוש של מאקרו אחד בתוך מאקרו אחר</p>
	<p>סיכום לגבי המאקרו: <u>נושאים שנותרו להניח ולא צריך לבדוק:</u> (1) הגדרת מאקרו תבוא לפני התייחסות למאקרו (2) בכל הגדרת מאקרו יופיע בסוף סיומת המאקרו (endmacro) (3) אין שגיאות בגוף המאקרו (את השגיאות, אם יש, מגלים בשלבים הבאים)</p> <p><u>נושאים שיש לבדוק:</u> (1) שם המאקרו תקין (אינו שם הוראה וכדו..) (2) בשורת ההגדרה ובשורת הסיום אין תווים נוספים אם יש שגיאה בשלב פרישת המאקרו - לא ניתן ליצר קובץ מקור מורחב. כתוצאה מכך, אי אפשר לעבור לשלבים הבאים ויש לסיים את התוכנית ולעבור לקובץ המקור הבא (אם קיים).</p>
	הבהרות סופיות של מיכל הרכזת