

Assembly program 01		
;Show a string		
.MODEL	small	1
.STACK	100h	2
.DATA		3
msg1	db 13,10,'Hello World',13,10,'\$'	4
msg2	db 13,10,'Hit any key to exit',13,10,'\$'	5
.CODE		7
mov AX,@data		8
mov DS,AX		9
		10
lea DX,msg1 ; show msg1 on screen		11
mov AH,09h		12
int 21h		13
exit:	lea DX,msg2 ; show msg2 on screen	14
mov AH,09h		15
int 21h		16
mov AH,01h ; read a char		17
int 21h		18
mov AH,4Ch ; end program		19
int 21h		20
END		

Assembly program 02		
;Accept a character and show it		
.MODEL small		1
.STACK 100h		2
.DATA		3
msg1	db 13,10,'Enter a character : \$'	4
msg2	db 13,10,'The character accepted is: \$'	5
msg3	db 13,10,'Hit any key to exit',13,10,'\$'	6
char	db 0	7
.CODE		8
mov AX, @data		9
mov DS, AX		10
lea DX,msg1	;Show msg1	11
mov AH,09h		12
int 21h		13
mov AH, 01h	;Read a character	14
int 21h		15
mov char,AL	;Save the character read	16
lea DX,msg2	;Show msg2	17
mov AH,09h		18
int 21h		19
mov DL, char	;Display the character	20
mov AH, 02h		21
int 21h		22
lea DX,msg3	;Show msg3 on screen	23
mov AH,09h		24
int 21h		25
mov AH, 01h	;Read a character	26
int 21h		27
exit: mov AH, 4Ch	;End program	28
int 21h		29
END		30

Assembly program 03		
;Accept a string up to 100 characters and show it		
.MODEL	small	1
.STACK	100h	2
.DATA		3
msg1	db 13,10,'Enter a string up to 100 characters : \$'	4
msg2	db 13,10,'The string accepted is: \$'	5
msg3	db 13,10,'Hit any key to exit',13,10,'\$'	6
str	db 100	7
strlen	db 0	8
strtxt	db 100 dup(0)	9
char	db 0	10
.CODE		11
	mov AX, @data	12
	mov DS, AX	13
	lea DX,msg1 ;Show msg1	14
	mov AH,09h	15
	int 21h	16
	lea DX,str ;Accept a string	17
	mov AH,0ah	18
	int 21h	19
	lea DX,msg2 ;Show msg2	20
	mov AH,09h	21
	int 21h	22
	mov AX,0 ;put \$ at the end of accepted string	23
	mov AL,strlen	24
	lea BX,strtxt	25
	add BX,AX	26
	mov [BX],'\$'	27
	lea DX,strtxt ;Show accepted string	28
	mov AH, 09h	29
	int 21h	30
	lea DX,msg3 ;Show msg3 on screen	31
	mov AH,09h	32
	int 21h	33
	mov AH, 01h ;Read a character	35
	int 21h	36
exit:	mov AH, 4Ch ;End program	37
	int 21h	38
END		39

Assembly program 04		
;Accept a character and show its previous and next		
.MODEL small		1
.STACK 100h		2
.DATA		3
msg1 db 13,10,'Enter a characters : \$'		4
msg2 db 13,10,'Hit any key to exit',13,10,'\$'		5
char db 0		6
crlf db 13,10,'\$'		7
.CODE		8
mov AX, @data		9
mov DS, AX		10
lea DX,msg1 ;Show msg1		11
mov AH,09h		12
int 21h		13
mov AH, 01h ;Read a character		14
int 21h		15
mov char,AL ;put the previous in char		16
dec char		17
lea DX,crlf ;New line		18
mov AH,09h		19
int 21h		20
mov AH,02h ;Show the previous		21
mov DL,char		22
int 21h		23
mov DL,'-' ;Show dash		24
int 21h		25
inc char ;Show current char		26
mov DL,char		27
int 21h		28
mov DL,'-' ;Show dash		29
int 21h		30
inc char ;Show next character		31
mov DL,char		32
int 21h		33
lea DX,msg2 ;Show msg3 on screen		34
mov AH,09h		35
int 21h		36
mov AH, 01h ;Read a character		37
int 21h		38
exit: mov AH, 4Ch ;End program		39
int 21h		40
END		41

Assembly program 05		
;Accept a character a-z and show its upper case		1
.MODEL small		2
.STACK 100h		3
.DATA		4
msg1	db 13,10,'Enter a characters a - z : \$'	5
msg2	db 13,10,'Invalid character \$'	6
msg3	db 13,10,'Hit any key to exit \$'	7
char	db 0	8
.CODE		9
mov AX, @data		10
mov DS, AX		11
lea DX,msg1	;Show msg1	12
mov AH,09h		13
int 21h		14
mov AH, 01h	;Read a character	15
int 21h		16
mov char,AL		17
cmp char,'a'	;If char < 'a' jump to invalid	18
jb invalid		19
cmp char,'z'	;If char > 'z' jump to invalid	20
ja invalid		21
mov DL,char	;Change letter to upper case	22
add DL,'A'		23
sub DL,'a'		24
mov AH,02H	;Show the uppercase letter	25
int 21h		26
jmp exit		27
invalid:	lea DX,msg2 ;Show Invalid character	28
	mov AH,09h	29
	int 21h	30
exit:	lea DX,msg3 ;Show msg3 on screen	31
	mov AH,09h	32
	int 21h	33
mov AH, 01h	;Read a character	34
int 21h		35
mov AH, 4Ch	;Return to operating system	36
	int 21h	37
END		38

הוראות אסמבלי

פקודה	C	S	Z	הערות
הוראות אריתמטיות				
1	MOV	opnd1, opnd2		משים את ערכו של opnd2 בתוך opnd1
2	ADD	opnd1, opnd2	✓	מוסיף ל opnd2 את ערכו של opnd1
3	SUB	opnd1, opnd2	✓	מחסר מ opnd1 את ערכו של opnd2
4	INC	opnd1	✓	מוסיף 1 לערכו של opnd1
5	DEC	opnd1	✓	מחסר 1 מערכו של opnd1
6	CMP	opnd1, opnd2	✓	משווה את ערכיהם של שני האופרנדים (בפועל הוא מבצע את פעולה 3 מבלי לשנות את ערכי האופרנדים)
7	MOV AL, 30 MOV BL, 4 MUL BL			מבצע פעולת כפל: $30 \times 4 = 120$ המכפלה AX = 120
7 מילה	MOV AX, 125 MOV BX, 200 MUL BX			מבצע פעולת כפל: $125 \times 200 = 25000$ המכפלה (DX AX) = 25000
8 בית	MOV AX, 205 MOV BL, 30 DIV BL			מבצע פעולת חילוק: $205 : 30 = 6$ (25) המנה AL = 6, השארית AH = 25
8 מילה	MOV DX, 0 MOV AX, 65012 MOV CX, 5000 DIV CX			מבצע פעולת חילוק של שני מספרים בני 16 ביטים $AX = (DX \ AX) / operand$ $65012 / 5000 = 13$ (12)
הוראות קפיצה				
9	JMP	label		מבצע קפיצה ללא תנאי
10	JZ / JE	label		מבצע קפיצה אם דגל האפס דלוק
11	JNZ / JNE	label		מבצע קפיצה אם דגל האפס מכובה
12	JS	label		מבצע קפיצה אם דגל הסימן דלוק
13	JNS	label		מבצע קפיצה אם דגל הסימן מכובה
14	JC	label		מבצע קפיצה אם דגל הנשא דלוק
15	JNC	label		מבצע קפיצה אם דגל הנשא מכובה
16	JA / JNBE	label		מבצע קפיצה אם לאחר ההשוואה
17	JAE / JNB	label		מבצע קפיצה אם לאחר ההשוואה
18	JB / JNAE	label		מבצע קפיצה אם לאחר ההשוואה
19	JBE / JNA	label		מבצע קפיצה אם לאחר ההשוואה

הוראות לוגיות					
20	NOT opnd1	0	✓	✓	הופך את ערכי הסיביות של opnd1
21	AND opnd1, opnd2	0	✓	✓	מפעיל את האופרטור AND על פי טבלת האמת. התוצאה ב opnd1
22	OR opnd1, opnd2	0	✓	✓	מפעיל את האופרטור OR על פי טבלת האמת. התוצאה ב opnd1
23	XOR opnd1, opnd2	0	✓	✓	מפעיל את האופרטור XOR על פי טבלת האמת. התוצאה ב opnd1
הוראות סיבוב					
24	SHL opnd1, num	✓	✓	✓	הזזת הסיביות של opnd1 שמאלה num פעמים
25	SHR opnd1, num	✓	✓	✓	הזזת הסיביות של opnd1 ימינה num פעמים
26	ROL opnd1, num	✓			הזזת הסיביות של opnd1 במעגל שמאלה num פעמים
27	ROR opnd1, num	✓			הזזת הסיביות של opnd1 במעגל ימינה num פעמים
28	RCL opnd1, num	✓			הזזת הסיביות של opnd1 במעגל שמאלה דרך הנשא num פעמים
29	RCR opnd1, num	✓			הזזת הסיביות של opnd1 במעגל ימינה דרך הנשא num פעמים

פסיקות		
קוד	פעולה	
mov AH, 01h int 21h	קריאת תו מהמקלדת התו ייקרא לתוך האוגר AL	30
mov DL, 'f' mov AH, 02h int 21h	הצגת התו 'f' על הצג	31
array db 200 actlen db 0 arrtext db 200 dup(0) lea DX, array mov AH, 0ah int 21h	קריאת מחרוזת באורך של עד 200 תווים לתוך מערך, המחרוזת תושם במערך לאחר הקשת Enter לאחר הקליטה יכיל actlen את מספר התווים שהוקשו בפועל	32
msg db 'what is your name?\$_' lea DX, msg mov AH, 09h int 21h	הצגת מחרוזת על הצג	33