University of Central Punjab
(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)
FACULTY OF INFORMATION TECHNOLOGY

Computer Organization and Assembly Language

Lab 10	
Topic	1. Video memory

Note: Ascii table is provided at the end.

PART 1

VIDEO MEMORY

Console Display:

Note : Each cell represents a word (2 byte).

Row 1,Col 1	**Row 1,Col 2**	**….**			**Row 1,Col 80**
Row 2,Col 1	**Row 2,Col 2**	**…**			**Row 2,Col 80**
…	**…**	**…**			**…**
…	**…**	**…**			**…**
…	**…**	**…**			**…**
…	**…**	**…**			**…**
…	**….**	**…**			**…**
…	**…**	**…**			**…**
Row 25,Col 1	**Row 25,Col 2**	**…**	**…**	**…**	**Row 25,Col 80**

; if you change the second byte, you can change the color of the character.

; character attribute is 8 bit value,

; high 4 bits set background color and low 4 bits set foreground color.

LET AX have 16 bits with character ‘A’ as a value byte and Brown background with white foreground color.

Blinking of the foreground color	Attribute byte							Value byte							
	Background			Foreground											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	1	1	1	0	1	0	0	0	0	0	1

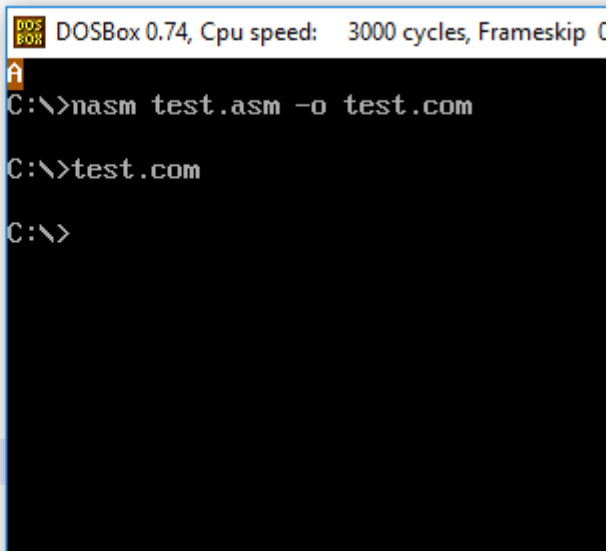
; hex	bin	color	
; 0	0000	black	possible background colors
; 1	0001	blue	
; 2	0010	green	
; 3	0011	cyan	
; 4	0100	red	
; 5	0101	magenta	
; 6	0110	brown	
; 7	0111	light gray	
; 8	1000	dark gray	possible foreground color
; 9	1001	light blue	
; a	1010	light green	
; b	1011	light cyan	
; c	1100	light red	
; d	1101	light magenta	
; e	1110	yellow	
; f	1111	white	

```
mov ax, 0xb800;
Mov es, ax;
mov di, 0;

mov ah, 0x6F;
mov al,0x41

Mov [es:di],ax;

mov ax,0x4c00
int 21h
```



Copy character array from one to another.

```
[org 0x100]
jmp start
data1 db 'Abcd,edfg,ijkl,mnopqr',0 ;this is zero means null.
data2: times 21 db 0
start:
mov si, data1
mov di, data2

l1:
mov al,[si]
mov [di],al
inc si
inc di
cmp al,0    ;comparing if the string is terminated or not.
jne l1

mov ax,0x4c00
int 21h
```

"To run code without debugging simply type test.com instead of afd test.com"

Type cls then enter before running the following codes.

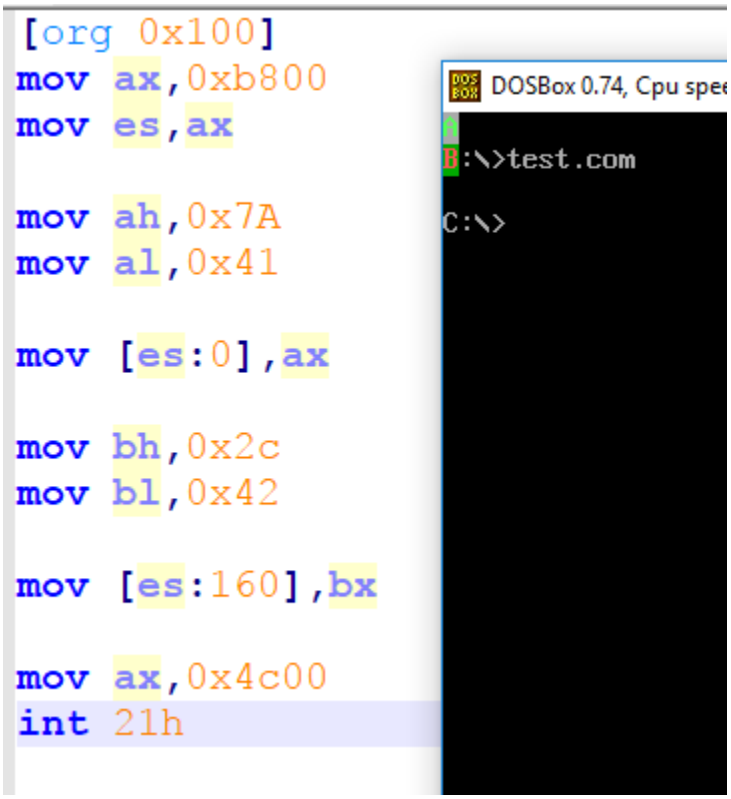
Display string on screen

```
[org 0x100]
jmp start
str1 db 'I am a student of University of Central Punjab',0
start:
mov ax, 0xb800;     ;segment address from where video memory starts.
Mov es, ax;
mov di, 0;          ;location on screen where we want to start displaying our string.
mov cx, 46;         ; string length, 11 characters.
mov si, str1;
mov ah, 0x1A;           ; Attribute byte for the characters to be displayed.
label:
Mov al, [si];           ;reading the characters in al.
Inc si              ; pointing to next character in string
Mov [es:di],ax;         ; printing message on the screen, whole register of size word is written at
Add di,2;
cmp cx,30
jne skip
change_blinking:
mov ah,0x9A
skip:
loop label

mov ax,0x4c00
int 21h
```


For example:

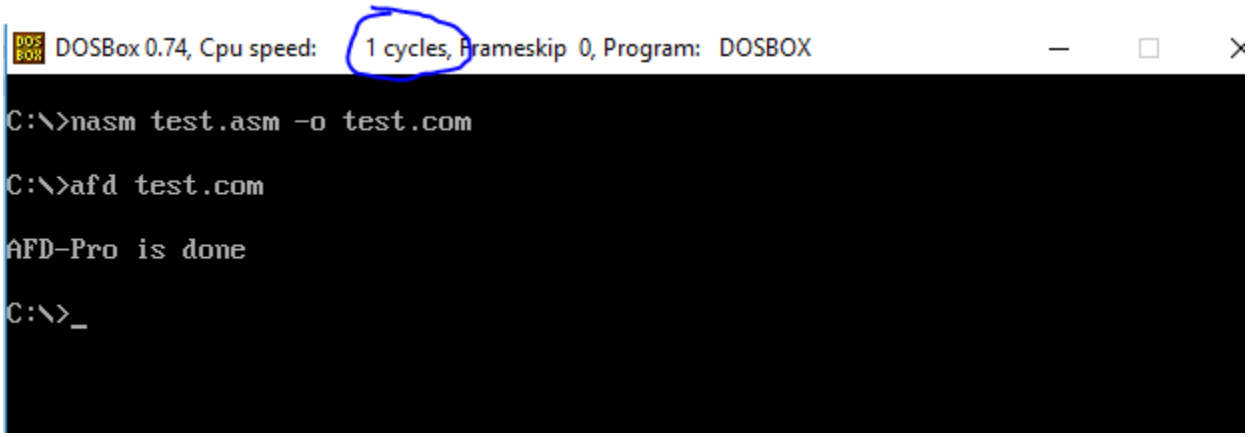
- Different attribute values of each word
- Different locations can be accessed for the display.



Display code which writes and clears the string from screen.

USE CTRL+F11 to reduce cycles / sec or CTRL+F12 to increase the speed of dosbox.

Slow down the speed of dosbox by press and hold ctrl and press F11 till 1 cycle




```
[org 0x100]
jmp start
str1 db 'HELLO WORLD'
start:
mov ax, 0xb800;
Mov es, ax;
mov di, 500;
mov cx, 11; ; string length, 11 characters.
mov si, str1;
mov ah, 0x1A;   ; Attribute byte, use any number
l1:
Mov al, [si];
Inc si; pointing to next character in string
Mov [es:di],ax; ; printing message on the screen;
Add di,2;
loop l1

mov cx, 2000;   ; total screen locations.
mov ax, 0x0720; Attribute byte (07) and (20h) ASCII for space character.
mov di, 0;  ; start from top left
l2:
Mov [es:di],ax; ; writing blank spaces on whole screen
Add di,2;
loop l2
mov ax,0x4c00
int 21h
```

University of Central Punjab

(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

ASCII CODES

HEX format

00:	null	20:	spa	40:	@	60:	`	80:	Ç	A0:	á	C0:	└	E0:	α
01:	☺	21:	!	41:	A	61:	a	81:	ü	A1:	í	C1:	┴	E1:	ß
02:	☻	22:	"	42:	B	62:	b	82:	é	A2:	ó	C2:	┬	E2:	Γ
03:	♥	23:	#	43:	C	63:	c	83:	â	A3:	ú	C3:	├	E3:	π
04:	♦	24:	$	44:	D	64:	d	84:	ä	A4:	ñ	C4:	─	E4:	Σ
05:	♣	25:	%	45:	E	65:	e	85:	à	A5:	Ñ	C5:	┼	E5:	σ
06:	♠	26:	&	46:	F	66:	f	86:	å	A6:	ª	C6:	╞	E6:	µ
07:	beep	27:	'	47:	G	67:	g	87:	ç	A7:	º	C7:	╟	E7:	τ
08:	back	28:	(	48:	H	68:	h	88:	ê	A8:	¿	C8:	╚	E8:	Φ
09:	tab	29:	)	49:	I	69:	i	89:	ë	A9:	⌐	C9:	╔	E9:	Θ
0A:	newl	2A:	*	4A:	J	6A:	j	8A:	è	AA:	¬	CA:	╩	EA:	Ω
0B:	♂	2B:	+	4B:	K	6B:	k	8B:	ï	AB:	½	CB:	╦	EB:	δ
0C:	♀	2C:	,	4C:	L	6C:	l	8C:	î	AC:	¼	CC:	╠	EC:	∞
0D:	cret	2D:	-	4D:	M	6D:	m	8D:	ì	AD:	¡	CD:	═	ED:	ø
0E:	♫	2E:	.	4E:	N	6E:	n	8E:	Ä	AE:	«	CE:	╬	EE:	ε
0F:	☼	2F:	/	4F:	O	6F:	o	8F:	Å	AF:	»	CF:	╧	EF:	∩
10:	►	30:	0	50:	P	70:	p	90:	É	B0:	░	D0:	╨	F0:	≡
11:	◄	31:	1	51:	Q	71:	q	91:	æ	B1:	▒	D1:	╤	F1:	±
12:	↕	32:	2	52:	R	72:	r	92:	Æ	B2:	▓	D2:	╥	F2:	≥
13:	‼	33:	3	53:	S	73:	s	93:	ô	B3:	│	D3:	╙	F3:	≤
14:	¶	34:	4	54:	T	74:	t	94:	ö	B4:	┤	D4:	╘	F4:	⌠
15:	§	35:	5	55:	U	75:	u	95:	ò	B5:	╡	D5:	╒	F5:	⌡
16:	▬	36:	6	56:	V	76:	v	96:	û	B6:	╢	D6:	╓	F6:	÷
17:	↨	37:	7	57:	W	77:	w	97:	ù	B7:	╖	D7:	╫	F7:	≈
18:	↑	38:	8	58:	X	78:	x	98:	ÿ	B8:	╕	D8:	╪	F8:	°
19:	↓	39:	9	59:	Y	79:	y	99:	Ö	B9:	╣	D9:	┘	F9:	∙
1A:	→	3A:	:	5A:	Z	7A:	z	9A:	Ü	BA:	║	DA:	┌	FA:	·
1B:	←	3B:	;	5B:	[	7B:	{	9B:	¢	BB:	╗	DB:	█	FB:	√
1C:	∟	3C:	<	5C:	\	7C:	¦	9C:	£	BC:	╝	DC:	▄	FC:	ⁿ
1D:	↔	3D:	=	5D:	]	7D:	}	9D:	¥	BD:	╜	DD:	▌	FD:	²
1E:	▲	3E:	>	5E:	^	7E:	~	9E:	₧	BE:	╛	DE:	▐	FE:	■
1F:	▼	3F:	?	5F:	_	7F:	⌂	9F:	ƒ	BF:	┐	DF:	▀	FF:	res

ASCII CODES

Decimal format

000:	null	032:	spa	064:	@	096:	`	128:	Ç	160:	á	192:	└	224:	α
001:	☺	033:	!	065:	A	097:	a	129:	ü	161:	í	193:	┴	225:	ß
002:	☻	034:	"	066:	B	098:	b	130:	é	162:	ó	194:	┬	226:	Γ
003:	♥	035:	#	067:	C	099:	c	131:	â	163:	ú	195:	├	227:	π
004:	♦	036:	$	068:	D	100:	d	132:	ä	164:	ñ	196:	─	228:	Σ
005:	♣	037:	%	069:	E	101:	e	133:	à	165:	Ñ	197:	┼	229:	σ
006:	♠	038:	&	070:	F	102:	f	134:	å	166:	ª	198:	╞	230:	µ
007:	beep	039:	'	071:	G	103:	g	135:	ç	167:	º	199:	╟	231:	τ
008:	back	040:	(	072:	H	104:	h	136:	ê	168:	¿	200:	╚	232:	Φ
009:	tab	041:	)	073:	I	105:	i	137:	ë	169:	⌐	201:	╔	233:	Θ
010:	newl	042:	*	074:	J	106:	j	138:	è	170:	¬	202:	╩	234:	Ω
011:	♂	043:	+	075:	K	107:	k	139:	ï	171:	½	203:	╦	235:	δ
012:	♀	044:	,	076:	L	108:	l	140:	î	172:	¼	204:	╠	236:	∞
013:	cret	045:	-	077:	M	109:	m	141:	ì	173:	¡	205:	═	237:	ø
014:	♫	046:	.	078:	N	110:	n	142:	Ä	174:	«	206:	╬	238:	ε
015:	☼	047:	/	079:	O	111:	o	143:	Å	175:	»	207:	╧	239:	∩
016:	►	048:	0	080:	P	112:	p	144:	É	176:	░	208:	╨	240:	≡
017:	◄	049:	1	081:	Q	113:	q	145:	æ	177:	▒	209:	╤	241:	±
018:	↕	050:	2	082:	R	114:	r	146:	Æ	178:	▓	210:	╥	242:	≥
019:	‼	051:	3	083:	S	115:	s	147:	ô	179:	│	211:	╙	243:	≤
020:	¶	052:	4	084:	T	116:	t	148:	ö	180:	┤	212:	╘	244:	⌠
021:	§	053:	5	085:	U	117:	u	149:	ò	181:	╡	213:	╒	245:	⌡
022:	▬	054:	6	086:	V	118:	v	150:	û	182:	╢	214:	╓	246:	÷
023:	↨	055:	7	087:	W	119:	w	151:	ù	183:	╖	215:	╫	247:	≈
024:	↑	056:	8	088:	X	120:	x	152:	ÿ	184:	╕	216:	╪	248:	°
025:	↓	057:	9	089:	Y	121:	y	153:	Ö	185:	╣	217:	┘	249:	∙
026:	→	058:	:	090:	Z	122:	z	154:	Ü	186:	║	218:	┌	250:	·
027:	←	059:	;	091:	[	123:	{	155:	¢	187:	╗	219:	█	251:	√
028:	∟	060:	<	092:	\	124:	¦	156:	£	188:	╝	220:	▄	252:	ⁿ
029:	↔	061:	=	093:	]	125:	}	157:	¥	189:	╜	221:	▌	253:	²
030:	▲	062:	>	094:	^	126:	~	158:	₧	190:	╛	222:	▐	254:	■
031:	▼	063:	?	095:	_	127:	⌂	159:	ƒ	191:	┐	223:	▀	255:	res

****Practice Tasks****

Task 1:

Write a program that will print “Hello world” on screen

Example:

HelloWorld HelloWorld

Task 2:

Write an assembly language program to display triangle of * on the screen as follows. Use loops to calculate the next screen location to display the next character.

i.e

if user give 5

*
**
***
****
*****

If user give 3

*
**
***

Task 3:

Write an assembly language program to display triangle of * on the screen as follows. Use loops to calculate the next screen location to display the next character.

i.e

if user give 4

*
***
*****
****

If user give 2

```
*
***
```

Task 4:

Write a program that will print “Hello world” on the half of screen and HelloWorld in reverse order on other half.

Example:

HelloWorld HelloWorld

dlroWolleH dlroWolleH