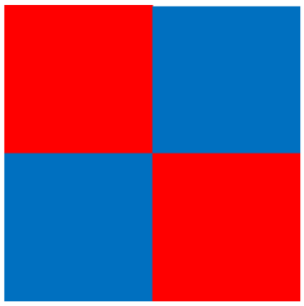
1) Write an assembly program, that will display the following pattern that covers the entire screen. User should be able to enter text only in the 4th block that is the bottom red block. Foreground Color should be white. No blinking.
Blocking function to return back to original mode is '*'



```
.model tiny
.data
strow1  db     00
stcol1  db     00
enrow1  db     12
encol1  db     40
strow2  db     00
stcol2  db     40
enrow2  db     12
encol2  db     80
strow3  db     12
stcol3  db     00
enrow3  db     24
encol3  db     40
strow4  db     12
stcol4  db     40
enrow4  db     24
encol4  db     80
prev    db     ?

.code
.startup
      mov    ah,0fh
      int    10h
      mov    prev,al
      mov    ah,0h
      mov    al,03
      int    10h
      mov    bh,0
      mov    dh,strow1
x1:   mov    dl,stcol1
      mov    ah,02h
      int    10h
      mov    ah,09h
      mov    al,20h
```

```
        mov     bl,47h
        mov     cx,40
        int     10h
        inc     dh
        cmp     dh,enrow1
        jnz     x1
        mov     dh,strow2
x2:     mov     dl,stcol2
        mov     ah,02h
        int     10h
        mov     al,20h
        mov     ah,09h
        mov     bl,17h
        mov     cx,40
        int     10h
        inc     dh
        cmp     dh,enrow2
        jnz     x2
        mov     dh,strow3
x3:     mov     dl,stcol3
        mov     ah,02h
        int     10h
        mov     al,20h
        mov     ah,09h
        mov     bl,17h
        mov     cx,40
        int     10h
        inc     dh
        cmp     dh,enrow3
        jnz     x3
        mov     dh,strow4
x4:     mov     dl,stcol4
        mov     ah,02h
        int     10h
        mov     ah,09h
        mov     al,20h
        mov     bl,47h
        mov     cx,40
        int     10h
        inc     dh
        cmp     dh,enrow4
        jnz     x4
        mov     bh,0
        mov     dh,strow4
        mov     dl,stcol4
        mov     cx,01
x6:     mov     ah,02h
        int     10h
        mov     ah,07h
        int     21h
        cmp     al,'*'
```

```
        jz      x5
        mov     ah,09h
        int     10h
        inc     dl
        cmp     dl,encol4
        jnz     x6
        mov     dl,stcol4
        inc     dh
        cmp     dh,enrow4
        jnz     x6
        mov     dh,strow4
        jmp     x6
x5:     mov     al,prev
        mov     ah,00h
        int     10h
.exit
End
```

2) Write an ALP that will display the first 10 characters in file 'mpi.txt' in the middle of the screen in the background and foreground specified by user as a command 'dis r w'. The command is interpreted as display 10 characters from file in red color(r) against a white background (w). The options for foreground & background can be:

r –red
b – black
B-blue
g-green
w-white

There is no blinking and all display is in low intensity. If the command is not dis nothing should be displayed. You can assume that file mpi.txt is already available in current directory.

Display blocking function is user press of '~' . When user presses the blocking key it should not be displayed on screen.

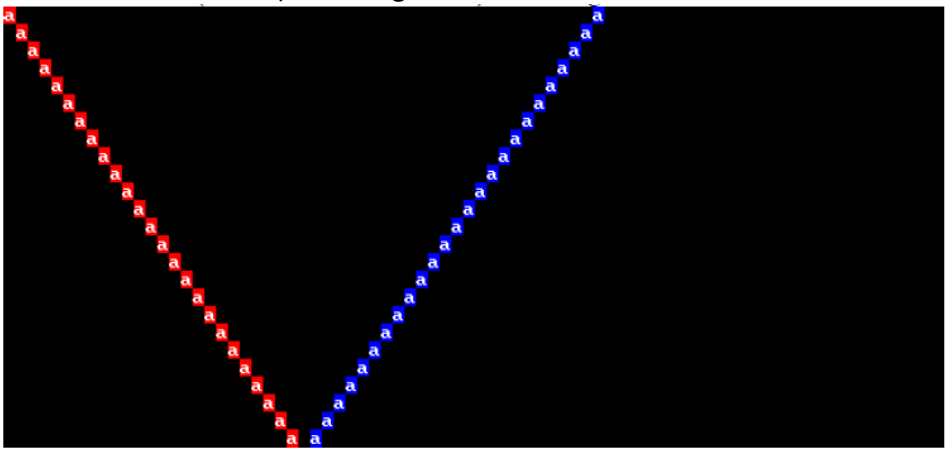
```
.model  tiny
.data
dat1    db      'dis'
dat2    db       8
dat3    db       ?
dat4    db       4 dup(0)
dat5    db       2 dup(0)
dat6    db       2 dup(0)
fil1    db      'mpi.txt',0
fdat1   db       10 dup(?)
atrb    db      'r','b','B','g','w'
atrb2   db       4h,0h,1h,2h,7h
att     db      ?
cno     db       36
.code
.startup
        mov     ah,0Ah
        lea     dx,dat2
```

```
        int     21h
        lea     si,dat1
        lea     di,dat4
        mov     cx,03h
x1:     cmpsb
        jnz     last1
        loop    x1
        mov     ah,3dh
        lea     dx,fil1
        mov     al,02
        int     21h
        mov     bx,ax
        mov     ah,3fh
        mov     cx,10
        lea     dx,fdat1
        int     21h
        mov     ah,3eh
        int     21h
        lea     si,dat5
        lea     di,atrb
        lea     bx,atrb2
c1:     mov     al,[si]
        cmp     al,[di]
        jz      x2
        inc     di
        inc     bx
        jmp     c1
x2:     mov     al,[bx]
        mov     att,al
        lea     si,dat6
        lea     di,atrb
        lea     bx,atrb2
c2:     mov     al,[si]
        cmp     al,[di]
        jz      x3
        inc     di
        inc     bx
        jmp     c2
x3:     mov     al,[bx]
        mov     cl,04
        rol     al,cl
        or      att,al
        mov     al,03h
        mov     ah,0
        int     10h
        lea     si,fdat1
        mov     di,10
x4:     mov     dh,12
        mov     dl,cno
        mov     bh,0
        mov     ah,02h
```

```
        int     10h
        lodsb
        mov     ah,09
        mov     cx,01h
        mov     bl,att
        mov     bh,0
        int     10h
        inc     cno
        dec     di
        jnz     x4
x5:     mov     ah,07h
        int     21h
        cmp     al,'~'
        jnz     x5
last1:
.exit
end
```

3) Write an ALP that will display the first character in file ‘mpi.txt’ in the format given below. If first character is ‘a’

Then display should be as follows. As you can see from figure entire row is occupied but not entire column (max column no will be 50). Blocking function is ‘?’



```
.model  tiny
.data
fil1    db      'mpi.txt',0
dat1    db      ?
cnt     db      25
cnt1    db      26
.code
.startup
        mov     ah,0
        mov     al,03
        int     10h
        mov     ah,3dh
        lea     dx,fil1
```

```
      mov     al,02
      int     21h
      mov     bx,ax
      mov     ah,3fh
      lea     dx,dat1
      mov     cx,1
      int     21h
      mov     bh,0
      mov     bl,11001111b
      mov     dl,0
      mov     dh,0
x1:   mov     ah,02
      int     10h
      mov     al,dat1
      mov     cx,1
      mov     ah,09h
      int     10h
      inc     dl
      inc     dh
      dec     cnt
      jnz     x1
      mov     bl,10011111b
x11:  mov     ah,02
      int     10h
      mov     al,dat1
      mov     cx,1
      mov     ah,09h
      int     10h
      inc     dl
      dec     dh
      dec     cnt1
      jnz     x11
x2:   mov     ah,07h
      int     21h
      cmp     al,'?'
      jnz     x2
      mov     ah,0
      mov     al,03
      int     10h
.exit
end
```

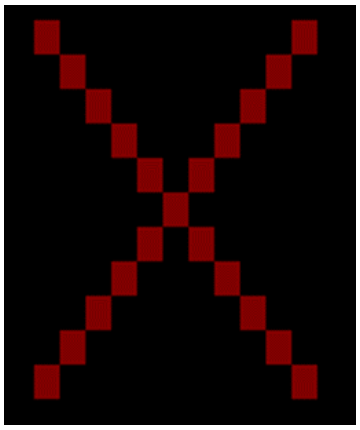
4) Write an ALP that will display a chessboard pattern on screen using text mode. The blocking function is user key press of '&'. User key press must not be visible on screen.

```
.model  tiny
.data
stcol   db      0
strow   db      0
encol   db      10
enrow   db      3
```

```
colsq   db      8
rowsq   db      8
prev    db      ?
inrow   db      10
incol   db      10
att     db      70h
.code
.startup
        mov     ah,0fh
        int     10h
        mov     prev,al
        mov     ah,00h
        mov     al,03h
        int     10h
        mov     bh,0
x4:     mov     dh,strow
x2:     mov     dl,stcol
        mov     ah,02h
        int     10h
        mov     al,20h
        mov     bl,att
        mov     cx,10
        mov     ah,09h
        int     10h
        inc     dh
        cmp     dh,enrow
        jnz     x2
        dec     colsq
        jz      x5
        mov     al,stcol
        add     al,incol
        mov     stcol,al
        mov     al,att
        cmp     al,70h
        jnz     x3
        mov     al,07h
        mov     att,al
        jmp     x4
x3:     mov     al,70h
        mov     att,al
        jmp     x4
x5:     dec     rowsq
        jz      x1
        mov     al,08
        mov     colsq,al
        mov     al,0
        mov     stcol,al
        add     al,10
        mov     encol,al
        mov     al,03
        add     strow,al
```

```
        mov     al,3
        add     enrow,al
        jmp     x4
x1:     mov     ah,07h
        int     21h
        cmp     al,'&'
        jnz     x1
        mov     al,prev
        mov     ah,0
        int     10h
.exit
end
```

5) Write ALP that will display the pattern shown in Figure above in the middle of the screen. The pattern should not blink. You will have to use text mode for this display. The pattern is a red cross (11 x11). The rest of the screen remains black. '?' should be used as blocking function.



```
.model  tiny
.data
orgdis  db      ?
stcol   db      35
strow   db      7
endcol  db      45
endrow  db      18
.code
.startup
        mov     ah,0fh
        int     10h
        mov     orgdis,al
        mov     ah,00
        mov     al,03
        int     10h
        mov     dh,strow
        mov     dl,stcol
x1:     mov     bh,0
        mov     ah,02
        int     10h
```

```
        mov     ah,09h
        mov     al,' '
        mov     bl,01000100b
        mov     cx,1
        int     10h
        inc     dl
        inc     dh
        cmp     dh,endrow
        jc      x1
        mov     dh,strow
        mov     dl,endcol
x3:     mov     bh,0
        mov     ah,02
        int     10h
        mov     ah,09h
        mov     al,' '
        mov     bl,01000100b
        mov     cx,1
        int     10h
        dec     dl
        inc     dh
        cmp     dh,endrow
        jc      x3
x2:     mov     ah,07
        int     21h
        cmp     al,'?'
        jnz     x2
        mov     ah,0
        mov     al,orgdis
        int     10h
.exit
end
```