

Set 1: Programe prin care se controleaza modul de afisare ecran a unui caracter colorat – a se vedea programul model de mai jos

**; Program care citeste de la tastatura modul cum sa se afiseze un caracter colorat pe ecran ;
; modifica si atributul de culoare pentru afisarea unui caracter pe ecran**

```
org 100h
; macrou pentru afisarea unui mesaj pe ecran
afisMsg macro
    mov AH,09h
    int 21h
endm

.data
msg0 db 'Tastati caracterul ce se doreste a fi afisat: ','$'
msgAp db 10,13,'Tastati nr de aparitii al caracterului ( 1 ... 9 ): ','$'
msgLin db 10,13,'Tastati linia unde sa apara caracterul ( 1 ... 9 ): ','$'
msgCol db 10,13,'Tastati coloana unde sa apara caracterul ( 1 ... 9 ): ','$'
msgErEr db 10,13, 'Repetati operatia !', '$'
msg1 db 10,13,'Tastati R, G sau B pentru a selecta culoarea de scriere a caracterului: ','$'

msgR db 10,13,' ROSIE ','$'
msgG db 10,13,' VERDE ','$'
msgB db 10,13,' ALBASTRA ','$'
msg2 db 'este culoarea selectata de dvs','$'

msgErCulScriere db 10,13,'Ceva nu a mers bine la selectarea culorii textului=>se patreaza valoarea anterioara', '$'
msgErCulFond db 10,13,'Ceva nu a mers bine la selectarea culorii fondului=>se patreaza valoarea anterioara', '$'

msgReia db 10,13,10,13,'Doriti sa reluati programul? y/n','$'

msg3 db 10,13,'Tastati R, G sau B pentru a selecta culoarea de background a caracterului: ','$'

caracter db ? ; caracterul ce se doreste a fi afisat se preia de la tastatura
nraparitii dw 0 ; nr de aparitii al caracterului se preia de la tastatura (1...9)
culoarea db 0F0h ; fond alb culoare de scriere , scris cu negru
linia db 0
coloana db 0

.code
start:
; secventa pentru curatarea ecranului
    mov ah, 0x00
    mov al, 0x03 ; text mode 80x25 16 colours
    int 0x10

; afisarea mesajului pentru preluarea de la tastatura a caracterului de afisat
    mov dx, offset msg0
    afisMsg
```

```
; preia caracterul de la tastatura si il salveaza in memorie
call preiaCar
mov [caracter], AL
```

```
; afisarea mesajului pentru preluarea de la tastatura a nr de aparitii a caracterului pe ecran
mov dx, offset msgAp
call verifica09
mov byte ptr [nraparitii], AL
```

```
; afisarea mesajului pentru preluarea de la tastatura a coloanei unde sa se afiseze caracterul
mov dx, offset msgCol
call verifica09
mov byte ptr [coloana], AL
```

```
; afisarea mesajului pentru preluarea de la tastatura a liniei unde sa se afiseze caracterul
mov dx, offset msgLin
call verifica09
mov byte ptr [linia], AL
```

```
; afisarea mesajului de modificare a culorii textului
mov dx, offset msg1
afisMsg
; preia o tasta de la tastatura
call preiaCar
```

```
; verifica tasta apasata si actualizeaza culoarea textului
mov BL, [culoarea]
cmp AL, 'R'
jne etGB
and BL, 0F0h ; se pune masca cu biti de 1 (se pastreaza) pe bitii culorii de fond
mov AL, 0Ch ; se impune culoarea de scriere ca fiind ROSIE
or BL, AL ; se combina cele 2 (culoarea de fond si cea de scriere)
mov [culoarea], BL ; salvam noua culoare in memorie
mov dx, offset msgR ; se va afisa un mesaj cu optiunea aleasa
jmp gataCulText ; a fost selectat R, deci nu mai verificam in continuare
```

```
; daca se ajunge aici, inseamna ca tasta apasata nu a fost R
```

```
etGB: cmp AL, 'G'
```

```
jne etB
```

```
and BL, 0F0h
```

```
mov AL, 0Ah
```

```
or BL, AL
```

```
mov [culoarea], BL
```

```
mov dx, offset msgG
```

```
jmp gataCulText
```

```
; daca se ajunge aici, inseamna ca tasta apasata nu a fost nici R, nici G
```

```
etB: cmp AL, 'B'
```

```
jne etEr
```

```
and BL, 0F0h
```

```
mov AL, 09h
```

```
or BL, AL
```

```
mov [culoarea], BL
```

```
    mov dx, offset msgB
    jmp gataCulText
; daca se ajunge aici, inseamna ca tasta apasata nu a fost nici R, nici G, nici B
etEr: mov DX, offset msgErCulScriere
    afisMsg
    jmp fond
; daca se ajunge aici, inseamna ca a fost una dintre R, G sau B
gataCulText: afisMsg
    mov DX, offset msg2 ; a doua jumtate a mesajului pentru afisarea optiunii alese pt culoarea de scriere
    afisMsg
```

fond: ; modificarea culorii backgroundului

```
; afisarea mesajului de modificare a culorii textului
mov dx, offset msg3
    afisMsg
; preia o tasta de la tastatura
call preiaCar
; verifica tasta apasata si actualizeaza culoarea textului
mov BL, [culoarea]
cmp AL, 'R'
jne etbGB
    and BL, 0Fh
    mov AL, 0C0h
    or BL, AL
    mov [culoarea], BL
    mov dx, offset msgR
    jmp gataCulFond
etbGB: cmp AL, 'G'
jne etbB
    and BL, 0Fh
    mov AL, 0A0h
    or BL, AL
    mov [culoarea], BL
    mov dx, offset msgG
    jmp gataCulFond
etbB: cmp AL, 'B'
jne etbEr
    and BL, 0Fh
    mov AL, 090h
    or BL, AL
    mov [culoarea], BL
    mov dx, offset msgB
    jmp gataCulFond
```

```
etbEr: mov DX, offset msgErCulFond
    afisMsg
    jmp aplica
```

```
gataCulFond: afisMsg
    mov DX, offset msg2
    afisMsg
```

```

; aici vom aplica toate modificarile dorite
aplica:
; secventa pentru curatarea ecranului
mov ah, 0x00
mov al, 0x03 ; text mode 80x25 16 colours
int 0x10

; pozitionare cursor pe ecran
mov DL,[coloana] ; coloana
mov DH,[linia] ; linia
mov AH,02h
int 10h ; cursor
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,[nraparitii]
int 10h ; afisarea colorata a caracterului

    mov DX, offset msgReia
    afisMsg

    call preiaCar
    cmp AL, 'n'
    jZ ies

    jmp start

ies:
ret
; procedura pentru preluarea de la tastatura a unui caracter
preiaCar proc
    mov Ah, 01h
    int 21h
    ret
preiaCar endp

verifica09 proc
dinNouNrAp: afisMsg

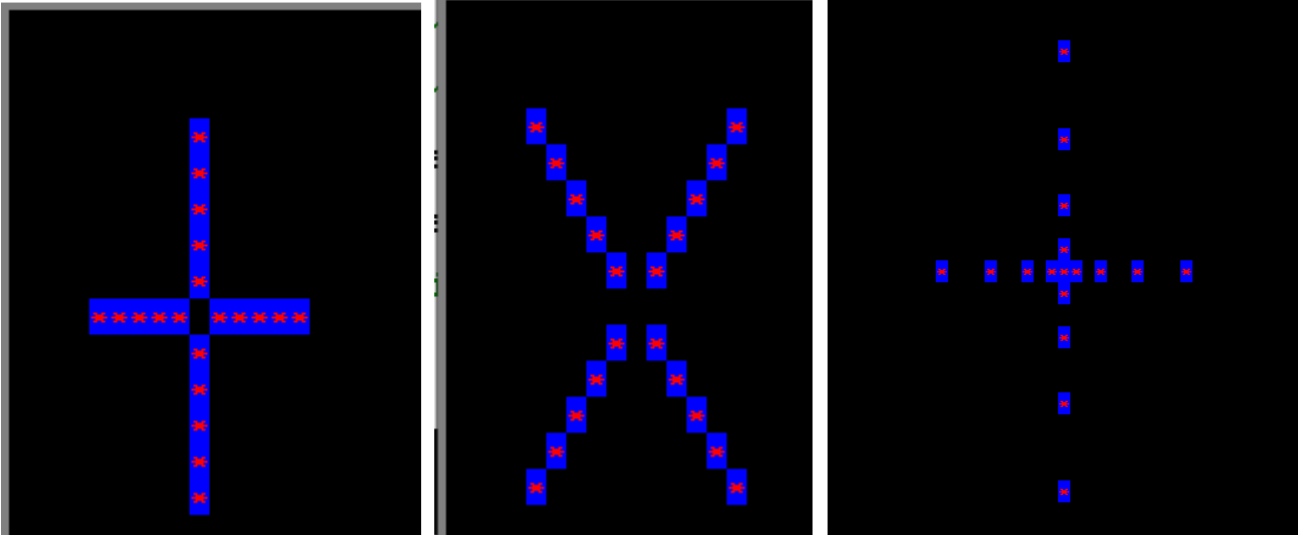
; preia nr de aparitii si il salveaza in memorie
call preiaCar
cmp AL,'1'
jl dinNouNrAp
cmp AL,'9'
jg dinNouNrAp
; daca se ajunge aici, s-a tastat corect un nr cuprins intre 1 si 9
sub AL, 30h
ret
verifica09 endp

```

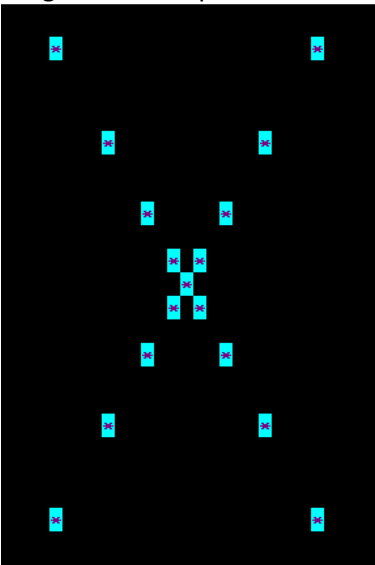
Modele de posibile cerinte la SET 1:

1. Sa se scrie un program care sa citeasca de la tastatura un caracter (de exemplu *), o pozitie pe ecran (linia si coloana) si apoi un numar (de exemplu 5). Programul va afisa pornind de la pozitia specificata un numar de 5 caractere inspre stanga, inspre dreapta, in sus si in jos (figura 1) sau pe diagonale (pentru diagonala se presupune ca linia/coloana este incrementata/decrementata cu 1 in fiecare pas) (figura 2), etc ...diverse efecte. Caracterul afisat va fi colorat rosu cu fond albastru. In program se vor afisa mesaje pentru fiecare interventie a utilizatorului si se va verifica tasta apasata de utilizator sa fie un numar cuprins intre 0 si 9.

Figura 3: programul porneste din mijlocul ecranului si afiseaza caracterul cu un salt progresiv sus-jos/stanga-dreapta.



Program care implementeaza modelul din figura de mai jos:



; Program care citeste de la tastatura un nr, porneste din mijlocul ecranului si merge progresiv pe diagonale pentru a afisa caracterul colorat

; (aici nu sunt afisate nici mesaje si nici nu se verifica daca tasta apasata este intre 0...9)

```
org 100h
```

```
; macrou pentru afisarea unui mesaj pe ecran
```

```
afisMsg macro
```

```
    mov AH,09h
```

```
    int 21h
```

```
endm
```

```
.data
```

```
caracter db ? ; caracterul ce se doreste a fi afisat se preia de la tastatura
```

nr dw 0 ; nr de aparitii al caracterului se preia de la tastatura (1...9)
culoarea db 0B5h ; culoare implicita
linia db 12
coloana db 40

```
.code  
start:  
; secventa pentru curatarea ecranului  
mov ah, 0x00  
mov al, 0x03 ; text mode 80x25 16 colours  
int 0x10
```

```
; preia caracterul de la tastatura si il salveaza in memorie  
call preiaCar  
mov [caracter], AL  
; preia numarul de la tastatura si il salveaza in memorie  
call preiaCar  
sub AL, 30h  
mov byte ptr [nr], AL
```

```
; aici vom aplica toate modificarile dorite  
aplica:  
; secventa pentru curatarea ecranului  
mov ah, 0x00  
mov al, 0x03 ; text mode 80x25 16 colours  
int 0x10
```

```
mov BL,0  
mov CX, [nr] ; nr de repetari ale buclei  
; punct de pornire  
mov DL,[coloana] ; coloana initiala  
mov DH,[linia] ; linia initiala
```

bucla:

```
push cx ; CX are rol dublu, de aceea trebuie salvat pe stiva  
; pozitionare cursor pe ecran  
add DL, BL ; adun la coloana un nr  
add DH, BL ; adun la linie un nr  
inc BL ; pentru cresterea progresiva, folosim un registru, de ex BL, care in bucla se incrementeaza  
mov AH,02h  
int 10h ; cursor
```

```
push bx ; BX are rol dublu, de aceea trebuie salvat pe stiva  
; afisare caracter  
mov BX, 1h ; specificam si pagina din memoria video  
mov AL, [caracter]  
mov AH,09h  
mov BL,[culoarea]  
mov CX,1  
int 10h ; afisarea colorata a caracterului in pozitia cursorului  
pop bx
```

```
pop cx
loop bucla
```

; repetam inca de 3 ori in celelalte directii

```
mov BL,0
mov CX, [nr]
mov DL,[coloana]
mov DH,[linia]
bucla2:
push cx
; pozitionare cursor pe ecran
```

sub DL, BL
add DH, BL
inc BL

```
mov AH,02h
int 10h ; cursor
push bx
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
pop bx
pop cx
loop bucla2
```

```
mov BL,0
mov CX, [nr]
mov DL,[coloana]
mov DH,[linia]
bucla3:
push cx
; pozitionare cursor pe ecran
```

add DL, BL
sub DH, BL
inc BL

```
mov AH,02h
int 10h ; cursor
push bx
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
pop bx
```

```
pop cx
loop bucla3
```

```
mov BL,0
mov CX, [nr]
mov DL,[coloana]
mov DH,[linia]
bucla4:
push cx
; pozitionare cursor pe ecran
```

```
sub DL, BL
sub DH, BL
inc BL
```

```
mov AH,02h
int 10h ; cursor
push bx
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
pop bx
pop cx
loop bucla4
```

```
ret
; procedura pentru preluarea de la tastatura a unui caracter
preiaCar proc
mov Ah, 01h
int 21h
ret
preiaCar endp
```

Deseneaza un dreptunghi, de o anumita lungime si latime , pornind de la o anumita pozitie pe ecran

```
; Program care deseneaza un dreptunghi pe ecran
org 100h
; macrou pentru afisarea unui mesaj pe ecran
afisMsg macro
mov AH,09h
int 21h
endm

.data
```


caracter db '*' ; caracterul ce se doreste a fi afisat se preia de la tastatura
lungime dw 15
latime dw 6
culoarea db 0B5h ; culoare implicita
linia db 12
coloana db 40

.code

start:

; secventa pentru curatarea ecranului
mov ah, 0x00
mov al, 0x03 ; text mode 80x25 16 colours
int 0x10

; aici vom aplica toate modificarile dorite
aplica:

; secventa pentru curatarea ecranului
mov ah, 0x00
mov al, 0x03 ; text mode 80x25 16 colours
int 0x10

;mov BL,0
mov CX, [lungime] ; nr de repetari ale buclei
; punct de pornire
mov DL,[coloana]
dec DL ; coloana initiala
mov DH,[linia] ; linia initiala

bucla:

push cx ; CX are rol dublu, de aceea trebuie salvat pe stiva
; pozitionare cursor pe ecran
add DL, 1 ; incrementez coloana (se creeaza latura lunga)
mov AH,02h
int 10h ; cursor

; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului in pozitia cursorului

pop cx
loop bucla

; repetam inca de 3 ori pt celelalte laturi

;mov BL,0
mov CX, [latime]
mov DL,[coloana]
mov DH,[linia]
dec DH
bucla2:

```
push cx
; pozitionare cursor pe ecran
```

```
inc DH ; cobor pe latime
```

```
mov AH,02h
int 10h ; cursor
```

```
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
```

```
pop cx
loop bucla2
```

```
mov CX, [lungime]
mov DL,[coloana] ; la coord de coloana a punctului de origine, adun o lungime
add DL, CL
dec DL
mov DH,[linia]
dec DH
mov CX, [latime]
```

```
bucla3:
push cx
; pozitionare cursor pe ecran
```

```
inc DH
```

```
mov AH,02h
int 10h ; cursor
```

```
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
```

```
pop cx
loop bucla3
```

```
mov CX, [latime]
mov DL,[coloana] ; la coord de coloana a punctului de origine, adun o lungime
```

```
mov DH,[linia]
add DH, CL
dec DH
```

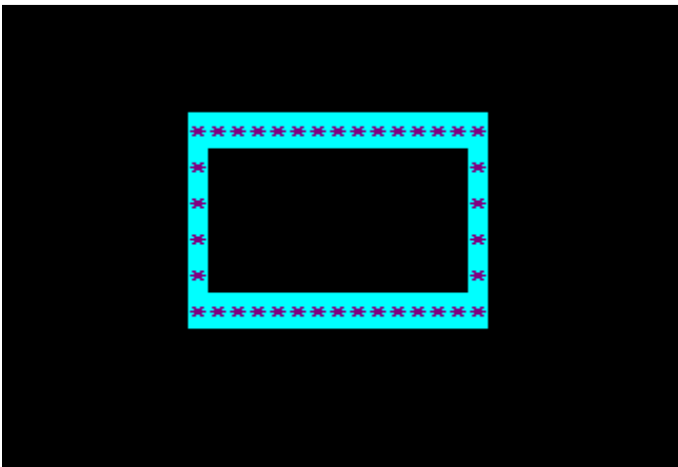
```
dec DL
mov CX, [lungime]
bucla4:
push cx
; pozitionare cursor pe ecran

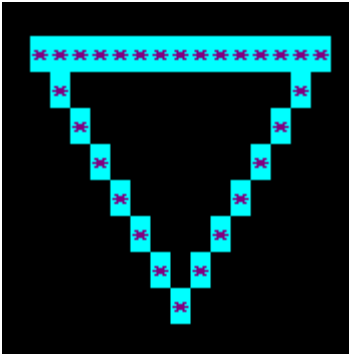
inc DL
mov AH,02h
int 10h ; cursor

; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului

pop cx
loop bucla4

ret
; procedura pentru preluarea de la tastatura a unui caracter
preiaCar proc
    mov Ah, 01h
    int 21h
    ret
preiaCar endp
```





; Program care deseneaza un dreptunghi pe ecran

org 100h

; macrou pentru afisarea unui mesaj pe ecran

afisMsg macro

mov AH,09h

int 21h

endm

.data

caracter db '*' ; caracterul ce se doreste a fi afisat se preia de la tastatura

lungime dw 15

latime dw 7

culoarea db 0B5h ; culoare implicita

linia db 12

coloana db 40

.code

start:

; secventa pentru curatarea ecranului

mov ah, 0x00

mov al, 0x03 ; text mode 80x25 16 colours

int 0x10

; aici vom aplica toate modificarile dorite

aplica:

; secventa pentru curatarea ecranului

mov ah, 0x00

mov al, 0x03 ; text mode 80x25 16 colours

int 0x10

;mov BL,0

mov CX, [lungime] ; nr de repetari ale buclei

; punct de pornire

mov DL,[coloana]

dec DL ; coloana initiala

mov DH,[linia] ; linia initiala

bucula:

push cx ; CX are rol dublu, de aceea trebuie salvat pe stiva

; pozitionare cursor pe ecran

add DL, 1 ; incrementez coloana (se creeaza latura lunga)

mov AH,02h

```
int 10h ; cursor
```

```
; afisare caracter
```

```
mov BX, 1h ; specificam si pagina din memoria video
```

```
mov AL, [caracter]
```

```
mov AH,09h
```

```
mov BL,[culoarea]
```

```
mov CX,1
```

```
int 10h ; afisarea colorata a caracterului in pozitia cursorului
```

```
pop cx
```

```
loop bucla
```

```
; repetam inca de 3 ori pt celelalte laturi
```

```
;mov BL,0
```

```
mov CX, [latime]
```

```
mov DL,[coloana]
```

```
mov DH,[linia]
```

```
;dec DH
```

```
bucla2:
```

```
push cx
```

```
; pozitionare cursor pe ecran
```

```
inc DH ; cobor pe latime
```

```
inc DL
```

```
mov AH,02h
```

```
int 10h ; cursor
```

```
; afisare caracter
```

```
mov BX, 1h ; specificam si pagina din memoria video
```

```
mov AL, [caracter]
```

```
mov AH,09h
```

```
mov BL,[culoarea]
```

```
mov CX,1
```

```
int 10h ; afisarea colorata a caracterului
```

```
pop cx
```

```
loop bucla2
```

```
mov CX, [lungime]
```

```
mov DL,[coloana] ; la coord de coloana a punctului de origine, adun o lungime
```

```
add DL, CL
```

```
dec DL
```

```
mov DH,[linia]
```

```
; dec DH
```

```
mov CX, [latime]
```

```
bucla3:
```

```
push cx
```

```
; pozitionare cursor pe ecran
```

```
inc DH
```

```
dec DL
```

```
mov AH,02h  
int 10h ; cursor
```

```
; afisare caracter  
mov BX, 1h ; specificam si pagina din memoria video  
mov AL, [caracter]  
mov AH,09h  
mov BL,[culoarea]  
mov CX,1  
int 10h ; afisarea colorata a caracterului  
pop cx  
loop bucla3
```

```
ret
```

```
; procedura pentru preluarea de la tastatura a unui caracter
```

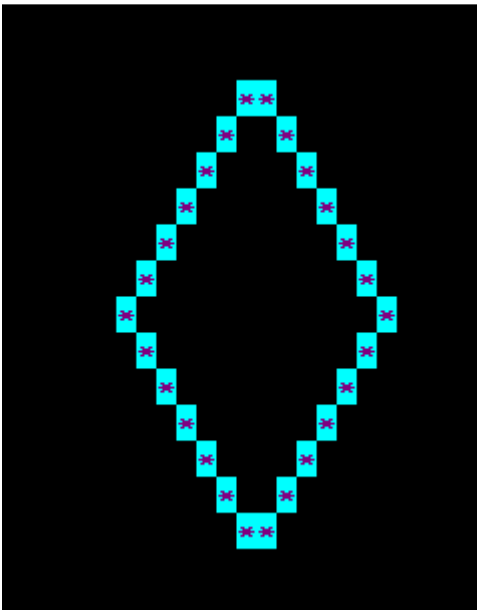
```
preiaCar proc
```

```
mov Ah, 01h
```

```
int 21h
```

```
ret
```

```
preiaCar endp
```



```
ROMB:
```

```
; Program care deseneaza un dreptunghi pe ecran
```

```
org 100h
```

```
; macrou pentru afisarea unui mesaj pe ecran
```

```
afisMsg macro
```

```
mov AH,09h
```

```
int 21h
```

```
endm
```

```
.data
caracter db '*' ; caracterul ce se doreste a fi afisat se preia de la tastatura
lungime dw 15
latime dw 7
culoarea db 0B5h ; culoare implicita
linia db 12
coloana db 40
```

```
.code
start:
; secventa pentru curatarea ecranului
mov ah, 0x00
mov al, 0x03 ; text mode 80x25 16 colours
int 0x10
```

```
; aici vom aplica toate modificarile dorite
aplica:
; secventa pentru curatarea ecranului
mov ah, 0x00
mov al, 0x03 ; text mode 80x25 16 colours
int 0x10
```

```
mov DL,[coloana]
mov DH,[linia]
mov CX,[latime]
dec DI
inc DH
bucla1:
push cx
; pozitionare cursor pe ecran
```

```
dec DH ; cobor pe latime
inc DL
```

```
mov AH,02h
int 10h ; cursor
```

```
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
```

```
pop cx
loop bucla1
```

```
mov DL,[coloana]
mov DH,[linia]
```

```
mov CX,[latime]
dec dl
dec dh
bucla2:
push cx
; pozitionare cursor pe ecran
```

```
inc DL
inc DH
```

```
mov AH,02h
int 10h ; cursor
```

```
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
pop cx
loop bucla2
```

```
mov CX, [lungime]
mov DL,[coloana]
add dl, cl
mov DH,[linia]
mov CX, [latime]
inc dh
inc dl
dec DI
dec DI
```

```
bucla3:
push cx
; pozitionare cursor pe ecran
```

```
dec DH ; cobor pe latime
dec DL
```

```
mov AH,02h
int 10h ; cursor
```

```
; afisare caracter
mov BX, 1h ; specificam si pagina din memoria video
mov AL, [caracter]
mov AH,09h
mov BL,[culoarea]
mov CX,1
int 10h ; afisarea colorata a caracterului
```



```
pop cx  
loop bucla3
```