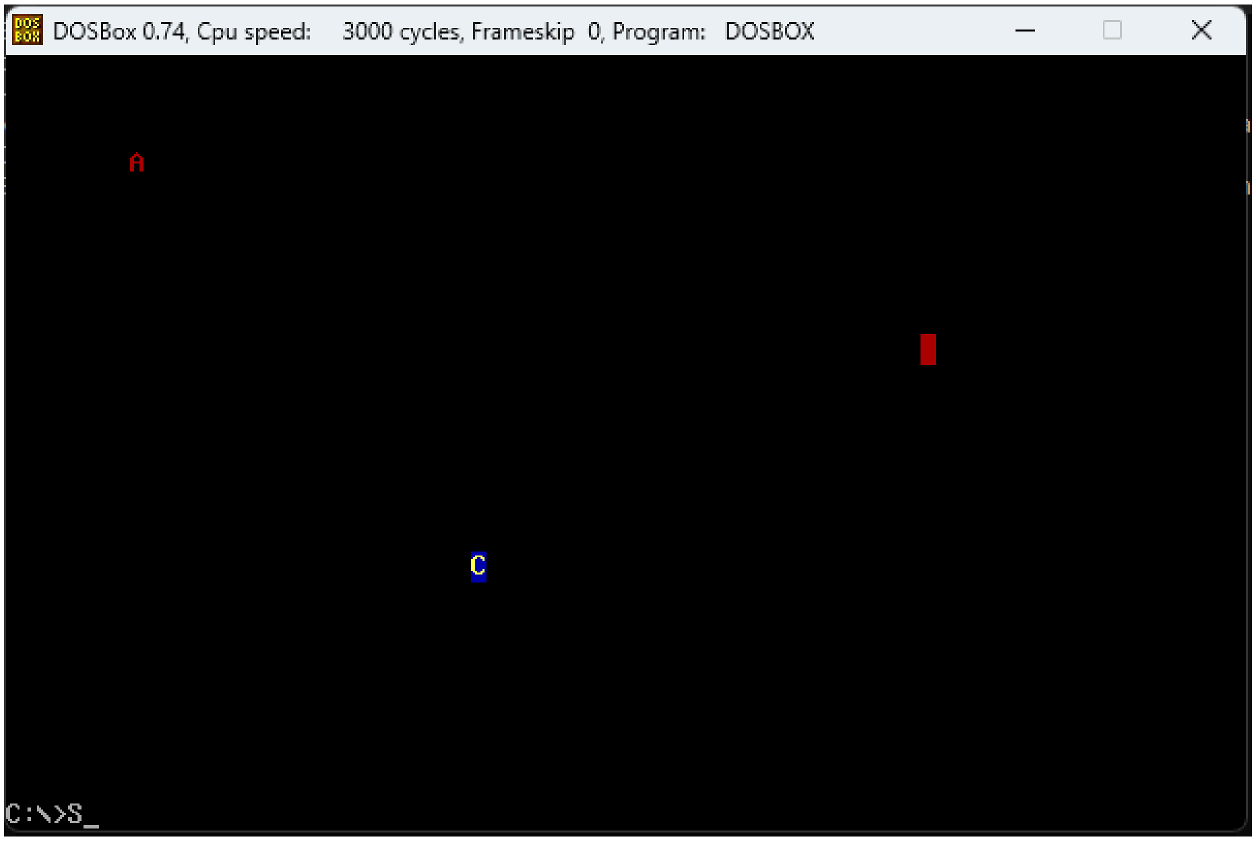
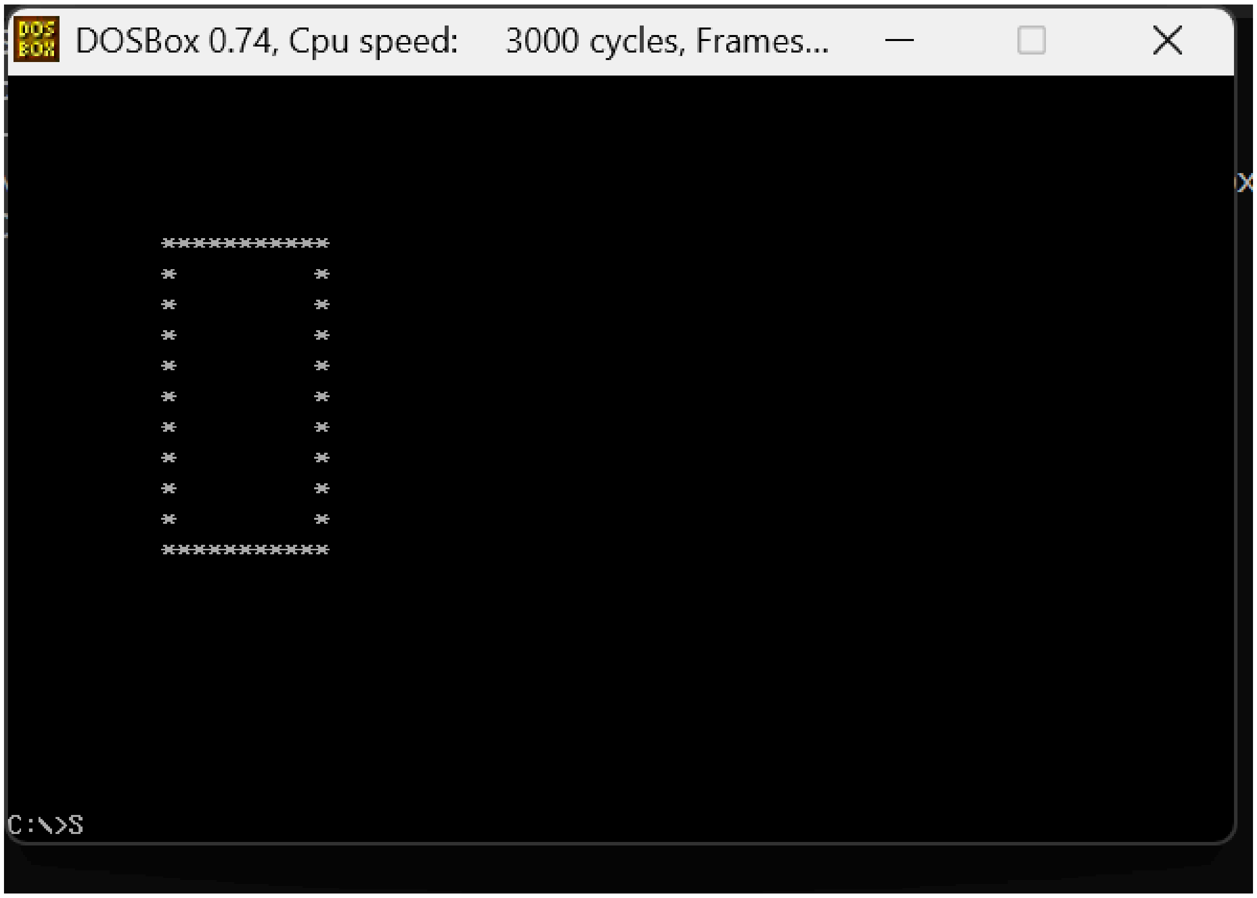
Q1.

```
[org 0x0100]
jmp start

char1 dw 'A'
char2 dw 'B'
char3 dw 'C'

clrscr:
    push es
    push ax
    push di
    mov ax, 0xb800
    mov es, ax
    mov di, 0

nextloc:
    mov word [es:di], 0x0720
    add di, 2
    cmp di, 4000
    jne nextloc

    pop di
    pop ax
    pop es
    ret

printchar:
    push bp
    mov bp, sp
    push es
    push ax
    push di

    mov ax, 0xb800
    mov es, ax
    mov di, [bp+6]
    mov al, [bp+4]
    mov ah, [bp+8]
    mov [es:di], ax

    pop di
    pop ax
    pop es
    pop bp
    ret 6

start:
    call clrscr

    ; A at position 656 - red foreground on black background - no blink
    push word 0x04
    push word 656
    mov ax, [char1]
    push ax
    call printchar

    ; B at position 1718 - bright green foreground on red background - blink
    push word 0xCA
    push word 1718
    mov ax, [char2]
    push ax
    call printchar

    ; C at position 2780 - yellow foreground on blue background - no blink
    push word 0x1E
    push word 2780
    mov ax, [char3]
    push ax
    call printchar

    mov ax, 0x4c00
    int 0x21
```


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
A

                                                                    █

                                   C

C:\>S_
```

Q2.

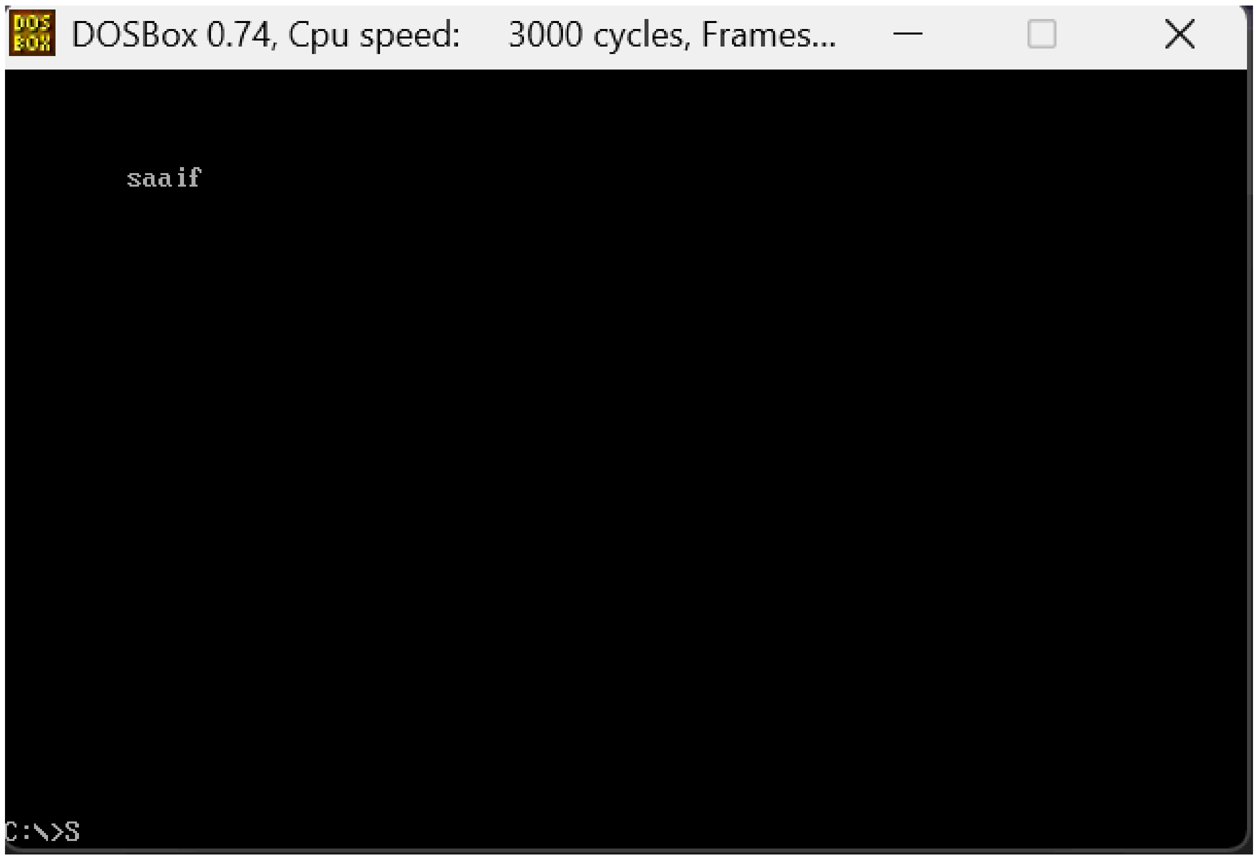


```
***********
*         *
*         *
*         *
*         *
*         *
*         *
*         *
*         *
*         *
***********




C:\>S
```

```
[org 0x0100]

start:
call clrscr
push 5 ; top row
push 10 ; left column
push 15 ; bottom row
push 20 ; right column
call drawrect
mov ax, 0x4C00
int 0x21

clrscr:
push es
push ax
push di
mov ax, 0xB800
mov es, ax
mov di, 0
```

```
nextloc:
mov word [es:di], 0x0720
add di, 2
cmp di, 4000
jne nextloc
pop di
pop ax
pop es
ret

drawrect:
push bp
mov bp, sp
push es
push ax
push bx
push cx
push dx
push di

mov ax, 0xB800
mov es, ax

mov ax, [bp+10] ; top row
mov bx, [bp+8] ; left column
mov cx, [bp+6] ; bottom row
mov dx, [bp+4] ; right column

push ax
mov di, ax
imul di, 80
add di, bx
shl di, 1
mov ax, 0x072A ; '*'

top_loop:
mov [es:di], ax
add di, 2
inc bx
cmp bx, dx
jle top_loop

pop ax
mov bx, [bp+8]
mov di, cx
imul di, 80
add di, bx
shl di, 1
mov ax, 0x072A

bottom_loop:
mov [es:di], ax
add di, 2
inc bx
cmp bx, dx
jle bottom_loop

mov bx, [bp+8] ; Left column
mov di, [bp+10]

left_loop:
push di
imul di, 80
add di, bx
shl di, 1
mov [es:di], ax ; Write '*'
pop di
inc di
cmp di, [bp+6]
jle left_loop

mov bx, [bp+4] ; Right column
mov di, [bp+10]

right_loop:
push di
imul di, 80
add di, bx
shl di, 1
mov [es:di], ax ; Write '*'
pop di
inc di
cmp di, [bp+6]
jle right_loop
pop di
pop dx
pop cx
pop bx
pop ax
pop es
pop bp
ret 8
```

Q3.


DOSBox 0.74, Cpu speed: 3000 cycles, Frames...

```
saaif

C:\>S
```

```
[org 0x0100]
jmp start
char1 db 's','a','a','i', 'f'

clrscr:
push es
push ax
push di
mov ax, 0xb800
mov es, ax
mov di, 0

nextloc:
mov word [es:di], 0x0720
add di, 2
cmp di, 4000
jne nextloc
pop di
pop ax
pop es
ret
```

```
printchar:
push bp
mov bp, sp
push es
push ax
push di

mov ax, 0xb800
mov es, ax
mov di, [bp+6] ; position
mov al, [bp+4] ; character
mov ah, [bp+8] ; attribute
mov [es:di], ax

pop di
pop ax
pop es
pop bp
ret 6

start:
call clrscr
mov cx, 5 ; number of characters
mov si, 0
mov dx, 656
print_loop:
mov al, [char1 + si]
push word 0x07
push word dx
push ax
call printchar
add dx, 2
inc si
loop print_loop

mov ax, 0x4C00
int 0x21
```